

Rec'd PCT/PTO 25 APR 2005

Application no/date: 1997-508852 [1995/ 8/19]  
 Date of request for examination: [1997/ 4/ ]  
 Accelerated examination ( )  
 Public disclosure no/date: 1997-512377 [1997/12/ 9]  
 Examined publication no/date (old law): [ ]  
 Registration no/date: [ ]  
 Examined publication date (present law): [ ]  
 PCT application no: PCT/EP95/003289  
 PCT publication no/date: WO97/007472 [1997/ 2/27]  
 Applicant: INTERNATL BUSINESS MACH CORP  
 Inventor: BANTATO KURUTO, PARUNKUTSUTO GEOFU, BEESHIYU ETSUKEHARUTO, RAISUTEN UDO  
 IPC: G06F 17/60, G06F 9/06, 530  
 FI: G06F 15/21, R G06F 9/06, 530U G05B 19/418, Z  
 G06F 17/60, 108, G06F 17/60, 164, G06F 17/60, 162C G06F 17/60, 104  
 F-Term: 5B049AA00, BB05, CC21, CC31, EE31, FF01, GG07, 3C100AA18, AA22, AA66, CC08, 5B076  
 EC09

Expanded classification: 454, 451

Fixed keyword:

Citation: [19, 1999. 9.30, 04 ] (04, , 94-18620 )

Title of invention: Process and method of and apparatus for *purojiekuto* administration  
 computer systems

Abstract: [PURPOSE]

It is put process control and *purojiekuto* administration together  
 by employing activity process *obujiekuto* as the data base which  
 process control and *purojiekuto* administration *niyotsute*  
 are common to.

[CONSTITUTION]

A central processing unit comprising necessary computing power functions  
 for center computer system 1030 for absorption process for the business  
 and a *purojiekuto* managerial system.

This computer includes activity process *obujiekuto* (WPO) 1001 in  
 data base.

This activity process *obujiekuto* (WPO) 1001 includes all static  
 definition and dynamic execution value for *purojiekuto* of the business.

Work place network server 1002 gives communication interface 1005 to individual  
*wakusuteshiyon* 1021-1026 of a user of this system 1030.

( Machine translation )

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表平9-512377

(43) 公表日 平成9年(1997)12月9日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 17/60  
9/06

識別記号

5 3 0

庁内整理番号

7925-5L  
8944-5B

F I

G 0 6 F 15/21  
9/06

R

5 3 0 U

審査請求 有 予備審査請求 未請求(全 73 頁)

(21) 出願番号 特願平9-508852  
(86) (22) 出願日 平成7年(1995)8月18日  
(85) 翻訳文提出日 平成9年(1997)4月18日  
(86) 国際出願番号 PCT/EP95/03289  
(87) 国際公開番号 WO97/07472  
(87) 国際公開日 平成9年(1997)2月27日  
(81) 指定国 EP(AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, M C, NL, PT, SE), JP, US

(71) 出願人 インターナショナル・ビジネス・マシーンズ・コーポレーション  
アメリカ合衆国10504、ニューヨーク州アーモンク、オールド・オーチャード・ロード (番地なし)  
(72) 発明者 バンタト、クルト  
オーストリア国ウイン、パウアーマルクト8  
(72) 発明者 パルンクット、ゲオフ  
イギリス国リミングトン、リリングトン、セブン・クロス 1  
(74) 代理人 弁理士 坂口 博 (外1名)

最終頁に続く

(54) 【発明の名称】 プロセスおよびプロジェクト管理コンピュータ・システム用の方法および装置

(57) 【要約】

本発明は、一般に、コンピュータ・ベースのプロセスおよびプロジェクト管理の技術分野と、プロセスおよびプロジェクト管理コンピュータ・システムとに関する。より具体的には、本発明は、共通の管理コンピュータ・システム上でプロセスおよびプロジェクト管理を行うための方法およびシステムに関する。現在、上質システムが求められているということは、作業プロセスのプロセスおよびプロジェクトの動的計画、管理、実行と、様々な役割でこのようなタスクを実行する人間とを相互依存的に統合する必要性を暗示している。個々のユーザの役割は作業プロセスに関する別々のビューを行使し、このような様々なビューは作業プロセスに関する部分的に重なる様々な機能を含む。本発明の概念は、相互依存的な役割モジュール作業プロセス環境用のシステム・プラットフォームを含む。作業プロセス・オブジェクト (WPO) (1001) は、プロセスおよびプロジェクト管理コンピュータ・システム (1030) のメモリ内で作成され、データ・ベース内に存在し、格納される。プロセスおよびプロジェクト管理に関するすべてのデータは前

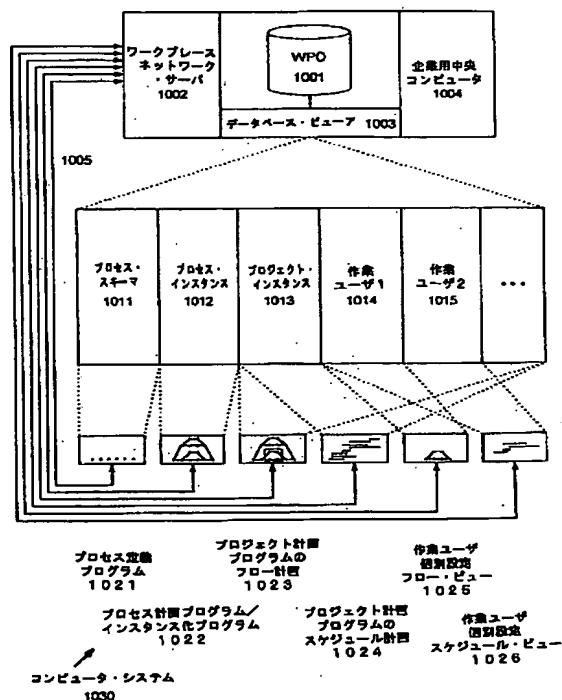


FIG. 10

**【特許請求の範囲】**

1. コンピュータ・システム（1030）上のプロセス管理およびプロジェクト管理のための方法であって、

前記プロセス管理が、デジタル・データによって表され、プロセス・スキーマ（101）から得られたプロセス・インスタンス（103）を含み、

前記プロジェクト管理が、デジタル・データによって表され、プロセス・インスタンス（103）から得られる（114）かまたはプロジェクト・スキーマ（102）から得られる（113）プロジェクト・インスタンス（104）を含み、

前記方法が、

作業プロセス・オブジェクト（1001）（WPO）が前記コンピュータ・システム（1030）のメモリ内に格納されたデータ・ベース内に存在する作業プロセス・オブジェクト（1001）（WPO）を作成するステップと、

前記プロセス管理および前記プロジェクト管理に関するデータを前記作業プロセス・オブジェクト（1001）（WPO）に報告するステップと、

前記プロセス管理と前記プロジェクト管理によって共通のデータ・ベースとして前記作業プロセス・オブジェクト（1001）（WPO）を使用するステップとを含むことを特徴とする方法。

2. 前記作業プロセス・オブジェクト（1001）（WPO）が有効グラフの構造を有し、

前記方法が、

第1のステージとして前記プロセス・スキーマ（101）から始まり、前記プロジェクト・インスタンス（104）の実行で終了する複数のステージごとに前記作業プロセス・オブジェクト（1001）（WPO）を進化させるステップと、

既存のステージの上にマッピングすることにより、前記ステージを構築するステップとをさらに含み、

前記既存のステージ上での前記マッピングがビューで達成され、前記ビューが

前記ステージのデータの指定を行うユーザ対話によって実現されることを特徴とする、請求項1に記載の方法。

3. 前記プロセス・スキーマ(101)に関するプロジェクト・スキーマ(102)をマッピングするために、定義済みプロセス・スキーマ(101)に関するタスク・エンベロープを定義するステップと、

前記プロセス・インスタンス(103)に関する前記プロジェクト・インスタンス(104)をマッピングするために、定義済みプロセス・インスタンス(103)に関するタスク・エンベロープを定義するステップとをさらに含むことを特徴とする、請求項1または2に記載の方法。

4. 定義域境界を使用して、前記作業プロセス・オブジェクト(1001)(WPO)の1つのステージ内の処理状態の

定義域を区切るステップと、

それに基づいて前記ステージ内で前記定義域境界を移動することができるような規則を提供する複数のビューを定義するステップとをさらに含み、

前記ビューが前記プロジェクトに関連するユーザ(1021、1022、1023、1024、1025、1026)用のユーザ対話によって表されることを特徴とする、請求項1ないし3のいずれかに記載の方法。

5. 複数の活動を有するバンドルと前記定義域とを前記バンドル用のパターン・グラフの上のデータのレベルによって表すステップをさらに含むことを特徴とする、請求項4に記載の方法。

6. 複数の活動を有する未拡張バンドル内のルート・スレッドの活動を含む水平エンベロープを使用するステップをさらに含み、

前記水平エンベロープが、前記ルート・スレッドのスレッド・インスタンス用の活動のすべてのインスタンスが同一タスクに属すことを確認し、

1つのバンドル内のスレッドの複数のインスタンス用の複数のタスクを定義するために垂直エンベロープを使用するステップをさらに含むことを特徴とする、請求項1ないし5のいずれかに記載の方法。

7. タスク・スタックを提供するステップであって、前記タスク・スタックがル

ープの拡張の結果得られる複数タスクの

スタックを表すステップをさらに含むことを特徴とする、請求項6に記載の方法。

8. 本来は前記プロセスで並行して実行可能な複数の活動用の隣接タスクのオーバーラップの程度の定義をサポートするステップをさらに含むことを特徴とする、請求項6または7に記載の方法。

9. プロセスおよびプロジェクト管理コンピュータ・システムのユーザ（1021、1022、1023、1024、1025、1026）に対してプロセスおよびプロジェクト管理の統合ビューを提示するステップをさらに含むことを特徴とする、請求項1ないし8のいずれかに記載の方法。

10. 請求項1ないし9のいずれかに記載のプロセスおよびプロジェクト管理のための方法を実行するコンピュータ・システム（1030）であって、前記コンピュータ・システムが、

プロセスおよびプロジェクト管理に共通して使用するデータ・ベースとして作業プロセス・オブジェクト（1001）（WPO）を格納するためのメモリと、

ユーザ（1021、1022、1023、1024、1025、1026）から前記コンピュータ・システム（1030）へ対話情報を入力するための入力手段（1002）と、

前記コンピュータ・システム（1030）から前記ユーザ（1021、1022、1023、1024、1025、1026）へ対話情報を送達するための出力手段（1005）

と、

前記入力手段（1002）と前記出力手段（1005）と前記メモリとの間の情報の流れを制御するためのプロセッサ（1004）とを含み、

前記プロセッサ（1004）が、前記プロセスおよびプロジェクト管理によるすべてのデータを前記作業プロセス・オブジェクト（1001）（WPO）に報告するための手段を含むことを特徴とする、コンピュータ・システム。

11. コンピュータ・プログラムを格納するデータ・キャリアであって、

前記コンピュータ・プログラムが、請求項 1 ないし 9 のいずれかに記載の方法  
を実行する請求項 10 に記載のコンピュータ・システム（1030）を制御する  
ことを特徴とする、データ・キャリア。

**【発明の詳細な説明】**

プロセスおよびプロジェクト管理コンピュータ・システム用の方法および装置

**技術分野**

本発明は、一般に、コンピュータ・ベースのプロセスおよびプロジェクト管理の技術分野、ならびにプロセスおよびプロジェクト管理コンピュータ・システムに関する。より詳細には、本発明は、共通の管理コンピュータ・システム上でプロセスおよびプロジェクト管理を行うための方法およびシステムに関する。

**背景技術**

新製品の設計、開発、製造を行うプロセスならびに既存製品の変更または調節を行うプロセスは、製品の品質を維持しまたはさらに向上させながら、所定のスケジュール内に最小コストで製品を市場に出すために、製品管理者とエンジニアに対して多くの難題を提示する。多くの企業は、従来の製品設計プロセスではこのようなニーズに対応するのに十分ではないと認識している。このような企業では、設計労力とともに、製造エンジニアリング、コスト・エンジニアリング、ロジスティック計画、調達、製造、サービス、サポートの早期関与を必要とする。さらに、このような企業では、設計、公

表、製造中の製品データの計画と管理も必要とする。

現代の管理ツールとしてのプロジェクト管理は、今世紀初頭、ヘンリー・ガント（Henry Gantt）が作業管理用の視覚的援助機能を開発したのを端初とする。いわゆるガント・チャートは、プロジェクト・スケジュールのグラフ表現であり、タスクの期間に比例した長さのバーとして各タスクを示すものである。

1972年11月28日に出願された、出願番号第54905/72号の「Process for converting activity on the arrow type network to an activity on the node」と題する英国特許第1397088号には、ノード型優先順位ネットワーク上の活動を処理可能なコンピュータ化プロジェクト管理システムが矢印型（arrow type）ネットワーク上の活動も処理できるようにするために、矢印型ネットワーク上の活動の外部表現をノード型優先順位ネットワーク上の活動の内部表現に変更するためのデジタル・コンピュータ・プロセスが記載されており

、前者の例は出願番号第54131/72号の英国特許第1375917号に開示されている。

G. P. フィッツパトリック (Fitzpatrick)、T. R. ヘインズ (Haynes)、W. R. スターレット (Sterrett) の論文「Transportation-Metaphor Workflow Status Display」(IBMテクニカル・ディスクロージャ・ブルテン、Vol. 35、No. 3、1992年8月、4～10ページ) には、ペーパー・バッチ処理からオンライン・ワークフロー処理への変更によって進行

中の作業の状況把握が非常に困難になると記載されている。用紙処理ツールにより、トランザクション経路が定義され、数人の内の1人が次の使用可能トランザクションを処理するために共通待ち行列からのトランザクションを要求することが可能になる。

D. J. スプーン (Spoon) の論文「Project Management Environment」(IBMテクニカル・ディスクロージャ・ブルテン、Vol. 32、No. a、1990年2月、250～254ページ) には、動作環境、データ要素、アプリケーション機能、プロセスを含む、プロセス管理環境が記載されている。

R. T. マルシャク (Marshak) の論文「IBM's FlowMark, Object-Oriented Workflow for Mission-Critical Applications」(Workgroup Computing Report(USA)、Vol. 17、No. 5、1994年、3～13ページ) には、ミッション・クリティカル生産プロセス・アプリケーションの開発および分配を目標とする、真のオブジェクト・モデル上に構築されたクライアント/サーバ製品としてのIBM FlowMarkのオブジェクトの特徴が記載されている。

H. A. インス (Inniss) およびJ. H. シェリダン (Sheridan) の論文「Workflow Management Based on an Object-Oriented Paradigm」(IBMテクニカル・ディスクロージャ・ブルテン、Vol. 37、No. 3、1994年3月、185ページ) には、カスタマイズおよび変更時のオブジェクト指向モデル化のその他の態様が記載されている。

F. レイマン (Leymann) およびD. ローラー (Roller) の論文「Business Proces



s Management with FlowMark」 (Digest of papers, Cat. No. 94CH3414-0、Spring COMPCON 94、1994年、230～234ページ) には、最新のコンピュータ・プロセス管理ツールであるIBM FlowMarkが記載されている。IBM FlowMarkの実施態様と同様に、IBM FlowMarkのメタ・モデルが提示されている。ビジネス・プロセスモデル化ならびにその実行用としてのIBM FlowMarkの可能性が論じられている。IBM FlowMarkという製品は様々なコンピュータ・プラットフォーム用に使用でき、IBM FlowMarkに関する文書はIBMのどの事業所でも入手可能である。

公報番号第0314596号、出願公開日が1989年5月3日の「Automated interface to project management tool」と題するヨーロッパ特許出願では、概念設計ツールの自動インタフェースのための方法が、最終製品を生産するための設計および製造の全労力を支援し改善するのに必要な早期製造関係情報を提供する。この概念設計ツールは、製品構造を構築し修正するために使用される。ユーザは、概念設計ツールを使用して、製品構造の項目別に製造詳細データを入力する。

1992年4月3日に出願され、1994年4月12日に特許が交付された「System and method for process modelling and project planning」と題する米国特許第5303170号には、変数としての時間、反復、資源タイプの柔軟な

処理を可能にするための方法が記載されている。たとえば、プロジェクト／プロセス・シミュレーション・ツールは、任意の活動を定義し、その活動を開始するのに必要な代替資源を定義し、その代替資源の可用性を決定し、代替資源の可用性に基づいて活動の持続期間を変化させる。別の態様によれば、モデル化プロセスは、ネットワーク内の活動の伝達関数をひとまとめに定義する複数の事前定義実行ステージを有するモデル化プログラムを供給すること、少なくとも1つの実行ステージを示すラベルがそれぞれに設けられている複数のユーザ供給プログラミング・ステートメントを定義すること、各ステートメントの直後に関連ラベルが示す実行ステージが続くようなユーザ供給プログラミング・ステートメントをモデル化プログラムに実行させることなどのために用意されたコンピュータであ

る。

本発明の「プロセス」という用語は、IBM FlowMark製品で定義可能なプロセスと非常に近い意味で使用する。これは、諸活動の順序付けを有向制御コネクタ、制御分割、結合によってモデル化する場合に有向グラフで諸活動のネットワークとしてプロセスを表すことができることを意味する。

「プロジェクト」という概念は、一般に、カレンダーによってしばしば定義される時間線上の諸活動の配置に関連する。各活動は、所定の持続期間を有するように計画され、資源、たとえば1人の人または複数の人からなるチームのメンバによって実行しなければならない。プロジェクト管理は、使用

可能な資源を使用し、諸活動の順序付け制約の範囲内を維持しながら、目標日を達成することに関する。したがって、プロセス管理者と協同しないプロジェクト管理では、通常、諸活動の順序付けの規則に関する暗黙の知識を使用する。一例として、家の建築を計画し管理する場合、屋根より先に地階を建築するなど、様々な活動を実行すべき順序は既知のものである。

#### 発明の目的

本発明の一目的は、プロセスおよびプロジェクト管理コンピュータ・システム用の方法および装置を提供することにある。

本発明の他の目的は、コンピュータ・システムに必要な最小限の計算能力および記憶容量だけで、プロジェクトまたはプロセスのいずれかに加えた変更が、対応するプロセスまたはプロジェクトに動的に反映され考慮されるようにすることにある。

本発明の他の目的は、プロセスおよびプロジェクト管理コンピュータ・システムのユーザに最大限の情報を提供することにある。

本開示の概念の主な目的は、プロセス管理は何を行うべきかとその論理的順序付けだけを定義し、プロジェクト管理がプロセスによって定義された順序付けの自由度の範囲内でカレンダー時間線内の諸活動の実行に必要な時間の側面にのみ寄

与するように、プロセス管理とプロジェクト管理を組み合わせることにある。こ

ここで使用する場合、プロセスは、任意の逐次化を行わずに論理的に許される最大限の並列性で行うべきすべての活動を定義し、プロジェクト管理は、そのプロジェクトについて要求されたスケジュールの範囲内でプロジェクトの実行のために使用可能な資源の制約を適用する。

#### 発明の概要

本発明の目的は、同封の独立クレイムに記載された特徴によって達成される。本発明のその他の構成および実施例は、対応する従属クレイムに開示されている。

現在、品質システムが求められているということは、作業プロセスのプロセスおよびプロジェクトの動的計画、管理、実行と、様々な役割でこのようなタスクを実行する人間とを相互依存的に統合する必要性を暗示している。個々のユーザの役割は、作業プロセスに対して異なるビュー（view）を働かせることであり、このような様々なビューは作業プロセスに関する部分的に重なる様々な機能を備える。

本発明の概念は、相互依存的な役割モジュール作業プロセス環境用のシステム・プラットフォームを含む。本発明の概念によれば、それぞれのビューは特定のワークスペースでサポートされ、特定の役割に対してその規則を実施しなければならない。作業プロセス情報モデルは、そのプロセスおよびプロジェクト計画モードとその実行の両面で作業プロセスを

表すデータ・ベース・オブジェクトの動的定義と使用をサポートする。本発明の概念によれば、1つのオブジェクトは、動的に移動する境界によって区切られたすべてのモードを同時にサポートする。ワークスペースの実施態様では、本発明のシステム内の役割ごとに全体的な規則を実施する。

独立クレイムである請求項1に記載の本発明は、先行技術における前述した制限および欠点を解消するものである。

データ・ベース内に存在する作業プロセス・オブジェクト（WPO）は、実行可能オブジェクトの定義とその結果行われる実行または解釈の両方について、プロセスおよびプロジェクト管理の統合概念として作業プロセスを表している。こ

のデータ・ベースは概念上は1つのデータ・ベースであるが、実際には1つのデータ・ベースと見なすことができる複数のデータ・ベースが存在する場合もある。プロセスおよびプロジェクト管理は、プロセス・スキーマおよびプロジェクト・スキーマと、プロセス・インスタンスおよびプロジェクト・インスタンスとを含み、これらのインスタンスはそれぞれのスキーマをインスタンス化することによって得られる。1つのプロセス・スキーマが複数のプロセス・インスタンスにインスタンス化することができるのと同様に、1つのプロジェクト・スキーマは複数のプロジェクト・インスタンスにインスタンス化することができる。各インスタンスは、それぞれのスキーマの特定の具体例と見なすことができる。プロジェクト・インスタンスは、プロジェクト・スキーマから得られ

るのと同様に、プロセス・スキーマから直接得ることもできる。

統合プロセスおよびプロジェクト管理によるすべてのデータが作業プロセス・オブジェクト（WPO）に報告されることは1つの利点である。というのは、これにより、その作業プロセス・オブジェクト（WPO）でプロセスおよびプロジェクト管理による1組の完全なデータ・セットを生成できるからである。さらに、この記憶概念によってコンピュータ・システム内の記憶ボリュームが低減されることも有利である。さらに、この記憶概念により、プロジェクトおよびプロセス管理コンピュータ・システム間の情報およびデータの交換のための管理オーバーヘッドが低減され、管理コンピュータ・システムの全体的なパフォーマンスが強化されることも有利である。さらに、作業プロセス・オブジェクト（WPO）が一般にプロセス管理とプロジェクト管理によってデータ・ベースとして使用されることも有利である。というのは、これにより、このような2つの管理目的間でデータ保全性が保証されるからである。

請求項2に記載の本発明の一実施例では、有向グラフの複数のステージからなる構造を有する作業プロセス・オブジェクト（WPO）がプロセス・スキーマから始まる複数のステージで開発され構築される。IBM FlowMark製品から分かるように、プロセス・スキーマはプロセス用のワークフロー・モデルの主要特性を備えている。このようなモデルは、作業プ

ロセス・オブジェクト（WPO）にも方向概念を導入する有向グラフの特性を備えている。作業プロセス・オブジェクト（WPO）のその他の追加データはいずれも、作業プロセス・オブジェクト（WPO）の最低ステージにおける基本プロセス・スキーマに関するビューによって連続マッピングを行った結果得られるものである。1つの基本プロセス・スキーマに基づいて複数のインスタンスをマッピングできるはずであるが、本発明の概念の説明では、1つの単一プロセス・スキーマから1つの実行可能プロジェクトおよびプロセスとその実際の実行へのマッピングとして、1つの作業プロセス・オブジェクト（WPO）を使用する。既存のステージの上にマッピングすることにより、複数のステージが構築される。このようなマッピングは、作業プロセス・オブジェクト（WPO）を目標ステージに入れるために必要なすべての詳細データの指定をサポートし、ユーザ対話によって実現されるビューで達成される。「ビュー」という用語は関係データ・ベースの観念で使用し、特定のビューが作業プロセス・オブジェクト（WPO）内の指定の情報にアクセスできるが、必ずしもすべての情報にアクセスできるわけではないことを意味する。ビューは、作業プロセス・オブジェクト（WPO）の完全な定義を制御し、様々なステージを作成し編集する。

作業プロセス・オブジェクト（WPO）は、各ビューが方向を持ったやり方でオブジェクトのステージを構築しなければならないという点で、有向オブジェクトである。この方向

を持ったやり方は、有向グラフによってIBM FlowMark製品などで表され、通常、左から右への方向を示す。本発明では、適切な表現だけの理由で、上から下への方向を想定し、図示する。したがって、各ビューは、上から下へオブジェクトのステージを構築しなければならない。作業プロセス・オブジェクト（WPO）のステージに関するビューを対話によってマッピングする処理は、必ず上から下へ行われる。すなわち、そのステージ上のマッピングがすでにどこまで進んだかを示す先行ステージ上を境界が移動することを意味する。

有向グラフの解釈は、ここでは上から下へであると想定する。IBM FlowMark製品から分かるように、制御分割後、並列スレッド内の諸活動に関して、上から下

への定義を一切与えることができない。並列スレッド内の諸活動間には隠れた同期がまったく存在しないと想定する。しかし、本発明の概念を拡張するとこのようなケースも含むことができるが、同期依存関係の明示的なモデル化が必要になると思われる。IBM FlowMark製品から分かるように結合の直前のすべての活動は、結合の上方向定義による。プロセスおよびブロック活動は、プロセスの観点では基本と見なされる。すなわち、それが含むスレッド内のサブプロセスまたはブロックに先行する活動および後続の活動以外のサブプロセスまたはブロック内の活動と、そのサブプロセスまたはブロック外の活動との間には、上方関係または下方関係を一切与えることができないことを意味する。

動的に変化する1つの作業プロセス・オブジェクト（WPO）がその様々な定義および実行時点での作業プロセスを表し、十分に定義された複数のビューがその作業プロセス・オブジェクト（WPO）のステージ間のマッピングをサポートすることは有利である。さらに、このように十分に定義されたビューにより、コンピュータ・システム内の記憶ボリュームが低減されることも有利である。さらに、ユーザ対話によってビューを連続実現することにより、プロセスおよびプロジェクト管理の実際に必要性および要件に適合させたビューの動的実現が可能になることも有利である。

請求項3に記載の本発明の他の実施例では、本発明の方法は、プロセス・スキーマまたはプロセス上にプロジェクト・スキーマまたはプロジェクトをマッピングするために、定義済みのプロセス・スキーマまたはプロセスに関するタスク・エンベロープを定義する。本発明で使用する「プロジェクト」という概念は、プロセスに関する所与の制約を行使し、カレンダーによって示される時間目盛上にプロセスをマッピングする。このため、プロジェクトは、まず、プロセス活動と、同一ユーザIDまたは同一人物に割り振るべき活動のエンベロープ・セットに関するタスクを定義する。タスク・エンベロープを識別することにより、プロセスに関してタスクが定義される。タスク・エンベロープの定義は、プロジェクト・スキーマの場合はプロセス・スキーマに関して、プロジェクトの場合はプロセスに関して行うことができる。

複合タスクは、実際にタスクの実行を開始する前に単純な実行可能タスクに改良しなければならない。単純タスクは、固有のユーザIDまたはチームIDに対して割り振る必要がある、実行可能な単純タスクは、1人の人または交代で作業を実行できる複数の人からなるチームのメンバに対して割り振ることができる。単純化のため、本開示ではチームについては説明しない。同じ理由で、タスクを実行するのは人間だけであり、プロセッサではないものとする。

1つのプロセス内の諸活動は、その活動に関連する資源によって実行すべき作業の諸要素を表している。本発明のコンテキストでは、活動タイプがプロセス・スキーマで定義されると想定する。プロセス・インスタンスでは、活動タイプ用のインスタンスを定義しなければならない。このような活動は単純タスクのコンテキストで実行しなければならない。このようなタスクもやはりそのプロジェクトのコンテキストで実行しなければならない。活動定義の詳細は、IBM FlowMark製品における諸活動について想定されているものである。活動タイプは、プロセス解釈時に後で活動インスタンスに改良することができ、拡張可能な活動という概念によって、後で挿入すべきプロセス・グラフの断片用の構文プレースホルダが得られる。

あらゆる基本活動にはプロセッサが1つずつ関連付けられているが、これは、処理ユニットについてはプロセッサ・タイプ、別個の処理ユニット、活動を実行する人のタイプにつ

いてはユーザの役割、識別される人についてはユーザIDにすることができる。IBM FlowMark製品の概念によるプロセス・スキーマとプロセス・インスタンスとの区別は、バンドル内、ならびに活動タイプのインスタンス内に位置する。バンドルはスキーマではグラフ内のスレッドを表し、概念上はそのスレッドの1つまたは複数の並列インスタンスを1つのプロセスで定義することができ、実際のスレッドをいくつ実行しなければならないかという正確な定義は、ほとんどの場合、実行中に動的にプロセス・インスタンスについて示される。

ここで検討するプロセスで定義される構文は、そのプロセスの純粋な論理でなければならない。すなわち、グラフ内の上記の活動の実行のプロセスに基づき、

しかも条件評価のためにこのような活動によって供給されるデータに基づき、どの条件下でどの活動を実行できるかである。言い換えると、定義されたプロセスは論理的に許される最大限の並列性を示し、その結果、活動の任意の逐次化を含まないところでは想定する。あるいは、さらに言い換えると、プロセス、処理ユニット、役割、およびユーザに関する資源制約の考慮は、一切プロセスにモデル化する必要がない。このような考慮は、プロジェクトにのみモデル化しなければならない。1組の規則によって最大限の並列性を示すためにグラフを検査できるかどうかについては、ここでは論じない。

請求項4に記載の本発明の他の実施例は、移動可能な定義域境界を備えた1つのステージ内の定義または処理状態の定

義域を含む。

その下で定義域境界を移動できる規則を提供するビューがユーザ対話によって表されることは有利である。これは、作業プロセス・オブジェクトを通して下方向への境界の移動に関連する改良がプロジェクト実行全体にわたって継続する動的プロセスであると認めている。さらに、本発明のデータ・アーキテクチャ概念のため、コンピュータ・システムの最小限の管理労力のみにより、これが達成されることも有利である。

請求項5および6に記載の本発明の他の実施例では、バンドルが複数レベルのデータによって表され、タスクを定義するために水平エンベロープと垂直エンベロープを使用する。「パターン・グラフ」という用語は、IBM FlowMarkの用語から分かるように、基本グラフの一種として使用する。また、「垂直」および「水平」エンベロープという用語は、有向グラフの実際の上から下へのビューに基づくものであり、そのビューを指している。したがって、有向グラフの左から右へのビューを設定する場合、これらの用語はその位置が変化する。

この実施例がプロセスおよびプロジェクト管理中にタスクの完全なピクチャを表すデータ・アーキテクチャをコンピュータ・システム内で提供することは有利である。

請求項6に記載の本発明の他の実施例では、タスク・スタックはループの拡張



の結果得られるタスクを表している。

このタスク・スタックが、ループの拡張によって作成されたタスクの実行のために上から下への垂直拡張を表すことができ、このループがIBM FlowMark製品により既知の様々なブロックであることは有利である。これにより、有利なことに、ループの内容の実行の各反復ごとに、デフォルト・ループ寸法によるかまたはループ反復情報が使用可能になったときに動的に、個別のタスク・エンベロープおよびタスクの作成が可能になる。タスク・スタック内のタスクは、通常のスケジューリング規則に従うので、オーバーラップするように定義することはできない。

請求項8に記載の本発明の他の実施例では、オーバーラップの程度の定義がサポートされる。

オーバーラップするものとして定義されたタスクはオーバーラップするスケジュールで定義できるので、これは有利である。さらに、プロセスの並列スレッドに関するエンベロープを定義するときに、各タスク・オーバーラップごとに先行／後続関係が定義され、この場合、その関係がプロセス定義によって暗示されないことも有利である。オーバーラップ・タスクの各対ごとに、後続タスクの活動化を可能にするために先行タスクのどの部分が実行済みでなければならないかは、その後続タスクについて定義される。オーバーラップする2つまたはそれ以上の先行タスクの場合、オーバーラップ条件は各先行タスクごとに別々に満たす必要がある。オーバーラップの尺度は、経過時間または費やした労力の時間、日数、月数、割合、ま

たはその他の可能な基準にすることができる。

請求項9に記載の本発明の他の実施例では、コンピュータ・システムのユーザに対して、プロセスおよびプロジェクト管理の統合ビューが提示される。

コンピュータ・システムのユーザが唯一のコンピュータ・システムからプロセスおよびプロジェクト管理の完全な情報を得ることができることは有利である。これは、プロセス管理とプロジェクト管理用の共通データ・ベースとして作業プ

プロセス・オブジェクト（WPO）を使用することによって達成される。さらに、プロセスおよびプロジェクト管理のための重複情報記憶を回避することにより、最小限のコンピュータ・システム資源だけでこれが達成されることも有利である。さらに、ユーザが選択可能な複数のビューのうちの1つで、プロセス内の可能な活動のコンテキストならびにこのような活動をエンベロープする活動化タスクからそのユーザに割り当てられた1組の活動をそのユーザに提示するような作業実行ワークスペースがプロジェクトに関連する人に対して提示されることも有利である。有利なことに、ユーザはこのワークスペースで自分に提示された活動についてプロセス・ビューと時間ビューとを選択することができる。

独立クレイムである請求項10に記載の本発明は、先行技術に関して前述した制限および欠点を解消するものである。

プロセスおよびプロジェクト管理によるすべてのデータが作業プロセス・オブジェクト（WPO）に報告されることは

1つの利点である。というのは、これにより、その管理用の完全なデータ・ベースが提供されるからである。さらに、プロセスおよびプロジェクト管理のための重複情報記憶を回避することにより、最小限のコンピュータ・システム資源だけでこれが達成されることも有利である。

請求項11に記載の本発明は、請求項10に記載のコンピュータ・システム上で請求項1ないし9に記載のコンピュータ・プログラムを媒介する可能性を提供する。

データ・キャリアにより、クライアントとサーバなどの複数のコンピュータ・システム上あるいはパーソナル・コンピュータ、ワークステーション・コンピュータ、メイン・フレーム・コンピュータなどの各種コンピュータ・プラットフォーム上でコンピュータ・プログラムの実行が可能になることは有利である。このデータ・キャリアは、機械、電気、磁気、または光学効果を使用してデジタルまたはアナログ方式で情報を格納することができる。

プロセスおよびプロジェクト管理の領域はまだ安定した用語の状態に達していないので、いくつかの用語については以下の説明で明確にする。

タスクの形式の作業は、資源割振りの機能によって人間に割り振られる。このような機能は最新技術を代表するものなので、ここではこれ以上説明しない。

作業割振りは、後でその作業を実行する人を識別するものである。作業割振りは、タスクに関する作業が割り振られた

人によって実際に実行できることを暗示しているわけではない。あらかじめ作業を人に割り当てておかなければ、その人は作業を開始することができない。

作業が任意の人に事前に割り振られ、その事前に定義されたスケジュールリング条件を満足したためにその作業を活動として含む特定のタスクが実行のために活動化されている場合、実際の実行のために作業がその人に割り当てられる。

作業実行は、プロセスにモデル化され、タスク内にエンベロープされ、タスクに関してプロジェクトによって割り振られ、実行資源に割り当てられた、作業活動の実際の実行である。人またはプロセッサなどの実行資源は、プロセス論理のためにその活動が実行のために使用可能になっていると同時にそのエンベロープ内の活動を含むタスクが活動化されている場合にプロセスおよびプロジェクト解釈の時点で割り当てられる。その時点で作業は人間に割り当てられる。活動開始とタスク開始ならびに活動終了とタスク終了の両方により、作業プロセス・オブジェクト（WPO）に記録すべき状況およびその他の情報が生成される。

本発明では、プロジェクト内の最後のタスクの実行によりプロセス・スキーマからのすべてのデータを記録する作業プロセス・オブジェクト（WPO）を開示する。この作業プロセス・オブジェクト（WPO）は、活動ノードから構築され、制御コネクタ、分割、結合によってリンクされる、有向グラフの概念上の基本構造を有する。この作業プロセス・オブジ

ェクト（WPO）内では、グラフ内の諸活動の個々の属性に関連するいくつかのサブオブジェクトを作成することができる。このような作業プロセス・オブジェクトのアーキテクチャと、その動的挙動に関する規則および手順の一部により、本出願における発明の態様が表される。

ビュー内の対話および機能は、作業プロセス・オブジェクトの1つまたは複数

の定義域を別の定義域にマッピングする。概念上、このような定義域は、特定の定義域用の複数レベルの属性を含む活動タイプのセット、ならびにタスク・サブオブジェクトなどのための定義域レベルに関連する追加のサブオブジェクトと見なすことができる。ただし、プロセス・スキーマのグラフはすべてのオーバーレイ定義域用の基本構造を含むことに留意されたい。IBM FlowMark製品の1概念であるバンドルの場合、その構造内の一部のスレッドについては、複数のインスタンスがそのスレッドのルートに関連付けられている。各インスタンスは、ルート・スレッドのコピーにより一部の定義域特性を継承し、ビューとそれに関連する対話により通常通り他の属性を獲得する。IBM FlowMark製品のプロセス活動の観念におけるブロックおよびサブブロックについては、本発明の態様にさらに貢献することなく、作業プロセス・オブジェクト（WPO）にさらに複雑さが持ち込まれるはずである。

ビューの対話により定義域の境界が移動し、下のステージの空または部分的にのみ開発した区分から上のステージの完

了区分を分離する。一般に、ビューと対話はこの概念に従い、定義域境界を上から下に移動する。例外的なケースでは、前に完了した区分における再作業、再編集、エラー訂正が必要な場合、定義域境界を上方向にリセットしなければならない場合もある。完了した区分を宣言して定義域境界をどこに配置するかは、ビュー対話を実行する人による明示的な判断になる。これにより、境界位置に適合するかまたは追加の編集作業を必要とする整合性検査が起動される。境界の下では、ユーザ宣言境界より上の区分について完全に整合性検査する前に、境界の下の不完全区分について新しい編集活動を行うことができる。各定義域ごとに、プロセス・スキーマの基本層の基本活動インスタンスの上に、あるレベルの定義域固有属性が構築される。この層により、区分間の境界の位置も記録される。

プロセス・スキーマ・ビューは、空の作業プロセス・オブジェクトの上にプロセス・スキーマをマッピングする。ビュー対話はプロセス・スキーマを構築するが、これは以下のように見なすことができる。

ー 活動タイプ・ノードからなるグラフ

- 制御コネクタ
- 制御分割
- 制御結合

活動タイプは、以下のクラスにすることができる。

- 基本活動
- プロセス活動
- ブロック活動

これは、IBM FlowMark製品の同様の概念に従って行われる。ブロック活動もループに対応する変形として現れるが、これは以下に定義するように特殊な処理を必要とする。グラフの所与の部分はバンドルのルートとして指定することもできるが、その特性についても後述する。正しいプロセス・スキーマは上から下へ構築するものと想定しているが、まだ完了していないスキーマについては、「上から下」の規則を適用せずに作成中に増大し変更することもできる。プロセス・スキーマの完全に作成された区分だけが、下のプロセス・スキーマ完了境界を備えている。スキーマのどの区分を完了したと見なすかは、プロセス・スキーマを編集する人の明示的な判断になり、その結果、整合性検査と境界の位置決めが起動される。

この完了境界の下では、プロセス・スキーマ・ビューの新しい作成アクションにより、依然としてプロセス・スキーマ境界の外側にある追加の区分の構築を開始することができる。区分は任意のものなので、スキーマ全体を同時に編集してそのスキーマの下に境界を設定できるはずである。区分の場合、ビューの境界を下方向に動的に移動できるが、追加区分の場合、他のビュー機能がその境界の下にすでにオブジェクト・データを作成するという手法は、後続のすべてのビューに適用可能である。

プロジェクト・スキーマ・マップは、一部のタスク・エンベロープとタスク定義であるプロジェクト・スキーマを、プロセス・スキーマの上にマッピングする。ここでは、複合タスクと単純タスクという2つのタイプのタスクを区別する。

複合タスクは、あらかじめ単純タスクに改良しなければ、実際に実行することができない。境界プロセス・インスタンス／プロジェクト・インスタンス・グロスは、複合タスクを含む区分から作業プロセス・オブジェクトのプロセス区分を区切るものである。割り振られたタスク境界は、依然として複合タスクを含むものから単純タスクだけを含むオブジェクトの区分を分離する。このような境界はすでにプロジェクト・スキーマに存在する可能性があるが、プロジェクト・インスタンスのみに関連する。

複合タスクはプロジェクト内で今後の計画を開発するという目的に対応するもので、詳細はプロセスおよびプロジェクト改良の後続ステージのみで示すことができる。この目的のため、後続ステージで拡張可能なプロセス活動、すなわち、その手順に含まれるグラフによって置き換えられるプロセス活動である、IBM FlowMark概念の拡張も使用しなければならない。この場合も、このような拡張可能手順の内容は活動タイプ用の活動インスタンスの作成時に後で指定することができる。複合タスクはネストすることができる。

プロジェクト・スキーマは、プロセス・スキーマに関するデフォルト・タスク構造を定義するという目的に対応するが、

これはプロセス・スキーマとともに、複数プロジェクト用の複数インスタンスの作成に使用することができる。プロジェクト・スキーマ・ビューの規則は、プロジェクト・インスタンス・ビューの規則と同じなので、ここで説明する。プロジェクト・スキーマ・レベルからの情報は、これがそれぞれのビューで作成されたときにプロジェクト・レベルにコピーされる。プロセス・スキーマに関するデフォルト・プロジェクト・スキーマを提供する機構は、本開示における本発明の概念の1つを表している。

プロセス・インスタンス・ビューは、1つの実行インスタンスについて、使用するプロセス・スキーマに関する改良を必要とするプロセスの詳細を定義する。特に、スキーマ内の活動タイプについて具体的な活動を識別しなければならず、バンドルについては、バンドル・スレッドのうち、計画し実行しなければならないインスタンスの数を解決する必要がある。1つのバンドル内の各スレッドごと

に、そのオブジェクト用のグラフに仮想並列スレッドが設定される。ただし、バンドルは必ずバンドル分割で始まり、バンドル結合で終わることに留意されたい。プロセス・インスタンス・ビュー機能は、プロセス・スキーマ／プロセス・インスタンス境界を上から下へ移動する。上記のように、これは区分単位で行うことができる。

プロセス・インスタンスのすべての要素、すなわち、活動とバンドル用として、作業プロセス・オブジェクト内の要素

用のプロセス・スキーマ層の上にプロセス・インスタンス層が置かれる。このような各層は、層固有の属性を含む。区分を使用する場合、一時的な区分用の対応する副層が挿入されるが、境界に及ぶ指定が完了し検査されると、後で除去される。バンドルを表す方法は、作業プロセス・オブジェクトの本発明のアーキテクチャ全体に関する本発明の補助概念である。

このビューは2つのサブビューを含み、どちらのサブビューも本発明の主な概念を含んでいる。割振り済みタスク境界は、それらを活動化できる場合にすでに詳細まで改良された単純タスクのみを含む作業プロセス・モデルの区分を、依然として複合タスクが実行可能性に達する前に解決、改良、または割振りを必要とするようなオブジェクトの一部分から分離する。

タスク・エンベロープ・ビューは、プロセス・スキーマまたはプロセスに関するタスク・エンベロープあるいはプロジェクトに関する複合タスク・エンベロープを識別するものである。

タスク計画ビューとその対話は、エンベロープ化したタスクをスケジュール及び実行資源ならびに一部のタスク属性と関連付けるものである。

タスク・エンベロープ・ビューは、タスク・エンベロープをプロセス・スキーマまたはプロセス・インスタンスまたはプロジェクト・インスタンスのグラフの上にオーバーレーす

る。単純タスクは、基本活動と複合活動の周囲の閉じた非交差エンベロープを提供する。単純タスクを定義する場合、以下の規則が関連する。

ー 単純タスク・エンベロープは、1つまたは複数の入力制御コネクタを有することができ、これはいずれも、グラフ内の上記のナビゲーション条件による活動から発生し、エンベロープ内の活動に達しなければならない。入力制御コネクタは、介入するナビゲーション条件（変換、分割、結合）なしで、エンベロープ内の活動に直接接続しなければならない。

ー プロセス内のすべての活動をプロジェクト用のタスク内に囲む必要はない。プロジェクトは、現在のIBM FlowMarkなど、プロセス実行規則により使用可能になり、人に割り当てられた非管理活動も含むことができる。

ー 2つのタスク・エンベロープ間のコネクタにおけるすべてのナビゲーション条件は、このようなエンベロープ外に置かれる。ナビゲーションのための条件評価は自動であり、プロジェクト管理によるスケジューリングの対象にはならない。

ー 単純タスクは、同じ役割または同じ人にすべて割り当てられた活動だけを囲むことができる。すべての複合活動（プロセス活動、ブロック）の場合、これは保証しなければならないが、必ずしも十分に検査できるわけではない。このような場合、違反が発生した場合にプロジェクト計画プログラム向けの例外を提起するような動的タスク割当て制御を含める必要がある。

ー 分割条件に従うタスクのエンベロープ化は特殊なケースである。1つまたは複数であるが、「m個の出力からn個選択」のすべてではない分割が入るタスクは、複合タスクとして定義しなければならない。すべてのコネクタがそのタスクに入るか、またはその条件が「すべての出力を選択」という条件である場合のみ、単純タスク用にエンベロープを定義することができる。ただし、IBM FlowMarkのいわゆる「誤伝播」により、上から下まですべての制御コネクタには真または偽の値が割り当てられ、未定義状態のままに維持できなくなることに留意されたい。

ー バンドルのエンベロープ化は特殊なケースである。この場合、まだ拡張していないバンドル用の水平エンベロープは、そのバンドルのルート・スレッド内の1つまたは複数の活動を含み、スレッド・インスタンス用のこれらの活動のすべ



てのインスタンスが同一単純タスクに属することを確認するか、または各スレッドあるいはこのようにエンベロープ化された場合はスレッドの各部分が個別タスクに拡張し、別々に実行資源に割り当てられることを暗示する垂直エンベロープ化を確認することができる。

ー 「オーバラップ」タイプのタスクは特殊なケースである。このようなタスクは、その実行時にオーバラップすることができる。2つまたはそれ以上のタスクは、1つまたは複数の第1のタスクとオーバラップすると定義することができる。第1のタスクとオーバラップ・タスクは隣接していなければ

ならず、これは、オーバラップ・タスクに入るすべての制御コネクタが1つまたは複数の第1のタスクから直接発生していなければならないことを意味する。このようなオーバラップは、いずれも同一分割に従うような2つまたはそれ以上の活動またはサブグラフ用に定義することができる。バンドルに関する水平タスクもスケジューリング時にオーバラップすることができる。水平タスクの場合、その順序付け規則は、以下に記載するタスク計画ビューによって個別に定義される。

ー ループ・ブロックの場合、2つのエンベロープ化モードが可能である。単純タスクによるループ・ブロックのエンベロープ化は、1つのタスク内でループのすべての反復を実行することを暗示する。タスク・スタックによるループ・ブロックのエンベロープ化は、それぞれが1つのレベルのタスク実行ごとの個別のタスクである、連続する単純タスクのスタックを暗示する。スタックの深さは、すでにプロジェクト・スキーマから得られたデフォルト値として送達されるか、または次の反復が始まる直前の場合にはそのプロジェクト内のその後の時点に得られる。未定義の反復因数を備えたタスク・スタックは、後述するように、実行可能タスク境界の位置決め用の複合タスクと見なされる。

ー いずれの場合も、タスク境界は互いに交差することができない。タスク境界は、ネスト・レベルの場合のみ接触することができる。その結果、たとえば、1つの活動を複数の単純タスクでエンベロープ化できなくなる。

1つの複合タスクは、1つまたは複数の複合タスクまたは単純タスクを含まなければならない。1つまたは複数の基本活動または複合活動あるいはその両方を含むことができる。複合タスクは、単純タスクへのその解決時に単純タスク用のコネクタに関する規則に従うように、すなわち、プロジェクト・インスタンス・ビュー対話の場合、単純タスクに関するコネクタ規則との競合の可能性を解消するために作業プロセスを編集する人を指導する適切なメッセージが作成されるように定義しなければならない。

定義による複合活動（ブロックおよびサブプロセス）は、1つの入力制御コネクタと1つの出力制御コネクタだけを備えている。これらが1つの単純タスクのエンベロープに含まれる場合、その内容は、上記のように単純タスクに関する規則によって実行しなければならない。これらが複合タスクに直接含まれる場合、それに含まれる活動は、エンベロープ化複合タスクに含まれる単純タスクによって別々にエンベロープ化することができる。

多くの場合、ほとんどの複合タスクは、そのより詳細な内容が定義済みになる前に定義されると想定しなければならない。したがって、ビュー対話のエラーは、改良によってこれらの詳細が改良済みになったときだけ、検出することができる。前述のように、各複合タスクごとに、まず、活動を定義しなければならない、それ自体は後でグラフ（8）で拡張することができる。このような活動を拡張する場合、それは、そ

れに関して定義された複合タスクとともに、作業プロセス・オブジェクトおよびそれに対応するグラフから消失する。その活動がプロセス活動である場合、それに関して定義された複合タスクは、複合タスクとしてオブジェクト内に維持される。したがって、複合タスクは、その後の改良のための一時的プレースホルダとして使用するか、または階層タスク構造内のより上位レベルのタスクを表すことができる。また、複合タスクは、オーバラップ・タスクとして指定することもできる。

エンベロープを描くことによって定義される各単純タスクまたは複合タスクごとに、プロジェクト内で固有のタスク名を定義しなければならない。一方、拡張

可能タスク用の一時レベルとしてまたは持続階層内のタスク用の永続レベルとして、タスク階層内のタスク・ネスト・レベル当たり1つのレベルを有する複合タスクの場合、このタスク名は、1つのタスク内に含まれるすべての活動のタスク関連層に関連付けられる。したがって、グラフの基本活動タイプの要素は、タスクの各ネスト・レベルごとにプロジェクト定義域内の1つの追加レベルを獲得し、各ネスト・レベルは、追加のレベルに含まれる情報を含む。そのグラフの活動タイプ要素はネスト以外の方法でタスクによって共用することはできないので、この機構は曖昧さなしに機能する。

このように識別された各タスクごとに、タスクに割り当てられた様々な属性を含む、作業プロセス・オブジェクトのサ

ブオブジェクトとしての個別オブジェクトが作成される。このような属性はタスク計画ビューで設定される。

プロセスは、プロジェクト内で何を行わなければならないかという論理を定義する。プロジェクト管理では、プロセス論理に違反せずに、使用可能な資源の制約内で所定のスケジュールを達成できるように、論理に関する制約を取り入れている。一般に、プロジェクト管理は、何らかのカレンダ線に沿ったタスクと活動の移動、ならびにプロジェクト計画の実行のための資源割当ての管理を含む。このような最適化活動は最新技術なので、ここではこれ以上説明しない。

ここで論じることは、プロジェクト管理アクションがプロセス・インスタンスによって定義された処理論理に違反しないようなやり方で、プロセス定義の存在がプロジェクト計画の編集、タスクのスケジューリング、資源の割振りをどのように制御できるかという新規の概念である。タスクは、それに関するスケジュール計画と資源割振りをプロジェクト内で適用しなければならないような活動のこのような集約を表すものとして定義されている。タスク計画ビュー用として、いくつかのサブビューが区別されている。

ー タスク役割またはユーザ・ビュー： 単純タスクのエンベロープを定義する場合、このエンベロープは、同じ役割または同じ人に割り当てられた活動用のみ描くことができる。役割は、そのタスクを実行するために割り振られたかまた

は割り当てられた人に対して後で決定される。この属性は、そ

のタスク用の特定のサブオブジェクトに割り当てられる。

ー タスク持続期間： プロジェクト・スキーマまたはプロジェクトを定義する場合、各タスクに持続期間を割り当てることができる。対応する値は、たとえば、デフォルト、推定、記録の計画、展開など、その計画の計画ライフ・サイクル属性によって記録することができる。

ー 対話式タスク・スケジュール定義： タスクには、開始日と終了日、あるいはタスクに関してすでに持続期間が存在する場合はそのうちの一方を割り当てることができる。開始日と終了日はどちらも、持続期間と同じ計画ライフ・サイクル属性に関連付けることができる。タスク・スケジュールの開始日または終了日が明示的または暗黙的に持続期間によって設定されている場合、開始日と終了日のいずれも、対応するタスク・エンベロープによって設定された上から下への規則に違反することができない。これは、すべての先行タスク（複数も可）が終了済みになるように計画されている場合のみ、タスクが開始するように計画できることを意味する。タスクはプロセス内の諸活動に関して定義されるので、したがって、活動実行の論理に対するタスクの依存が自動的に実施される。タスク開始日と終了日は、プロジェクト・スキーマ内のタスクについて定義することはできない。未管理の活動がタスク間に散在している場合、スケジュール検査のため、このような活動は無視される。

ー バッチ・タスク・スケジュール定義： 場合によっては、

タスク順序付けの整合性を検査せずに、複数のタスクに連続して上から下へスケジュール値を割り当てることが適切な場合もある。その場合、上から下へ分析して、非整合のメッセージ・リストが生成される。バッチ・スケジュール割当てと並行して、検出された第1の非整合に続くすべての変更をリセットするために使用可能な割当てログが生成される。

ー オーバラップ・タスク用のタスク・スケジュール定義： オーバラップするものとして定義されたタスクは、オーバラップ・スケジュールで定義することが

できる。プロセスの並列スレッドに関するエンベロープを定義する場合、各タスク・オーバーラップごとに先行／後続関係が定義されるが、この場合、それはプロセス定義によって暗示されない。オーバーラップ・タスクの各対ごとに、後続タスクについて、その後続タスクの活動化を可能にするために先行タスクのどの部分を実行しなければならないかが定義される。オーバーラップの尺度は、経過時間または費やした労力の時間、日数、月数、割合、または本開示に関連しないその他の可能な基準にすることができる。2つまたはそれ以上の第1のタスクがオーバーラップする場合、オーバーラップ条件はそれぞれの第1のタスクごとに個別に満たす必要がある。

ー タスク・スタック用のタスク・スケジュール定義： タスク・スタックは、IBM FlowMarkからの様々なブロックであるループを拡張することによって作成されたタスクの実行のための上から下への垂直拡張を表す。この場合、ループの内

容の実行の各反復ごとに、デフォルト・ループ寸法によるかまたはループ反復情報が使用可能になったときに動的に、個別のタスク・エンベロープおよびタスクが作成される。タスク・スタック内のタスクは、通常のスケジュール規則に従うので、オーバーラップするように定義することはできない。

ー 分割ナビゲーション条件により複数の可能なスレッド間で選択が行われる場合、ヌル・タスクを宣言し、非選択スレッドの実行を解消することができる。ワークフロー・マネージャであるIBM FlowMarkによる誤伝播は、休止スレッドにより真／偽値をプロモートするように取り計らう。上記で定義するように「m個の出力からn個選択」という分割に従うことができるのは複合タスクだけなので、実行のためにタスクを解決するか、またはタスクを宣言し、適用可能な場合にその後続タスクをヌルに宣言するかは、複合タスクを解決する計画プログラムの判断になり、その場合、スケジューリングおよび資源割振り機構に対しては無視される。ワークフロー・マネージャがその現行状態に関する制御コネクタの問合せのためにアプリケーション・プログラム・インタフェース（API）を用意している場合、より精巧なアーキテクチャであれば、ヌル割当ての定義および伝播を自動的に自動化する●●●もできる。●●●

ー タスクは、実行可能になるためには人に割り振らなければならない。このような割り振りは、資源管理において最新技術である資源割り振り方法によって行うことができる。

割り振り済みタスク境界は、すでに割り振られた実行可能な単純タスクだけを含む完全なプロジェクト・グラフのその区分を、複合タスクまたは未割り振りタスクを含む可能性のある割り振り済みタスク境界の下から区分するものである。単純タスクに従う未管理活動は、割り振り済みタスク境界内に含まれる。ヌル・タスクもその境界内に含まれる。この境界は、グラフが上部セクションと下部セクションに完全に区分されるように、そのグラフのすべてのコネクタと交差する1本の連続線でなければならない。

作業プロセスの実行が割り振り済みタスク境界に達すると、実行不能複合タスクまたは未割り振りタスクのためにプロジェクトが部分的に停止されたことを計画プログラムに通知する例外が提起される。

作業プロセスは原則として管理されるものと想定するが、これは定義済みタスクを含むが、未管理活動も含むことができる。上記で定義するように、割り振り済みタスク境界は、実行のために作業を割り当てることができるようにプロジェクトのどの部分を所定のレベルの詳細まで定義するかを動的に定義する。作業プロセス・オブジェクトを連続的に解釈する作業実行管理者の存在により、上記の規則で定義されるように、先行タスクが達した状態に基づいて、割り振り済みプロセス部分が識別され、その適切な活動化のためにタスク定義が分析されるものとしてここで想定する。

この実行機構は、先行タスクの完了のためにすでにタスク

を開始できることを分析し、その状況で計画プログラムを変更し、計画したスケジュールが達成されていない場合に例外条件を検出するように、すべての例外も監視する。このような監視活動はいずれも、最新技術のプロジェクト管理に多少なりとも存在する機能を表している。革新的な概念は、プロセス実行とプロジェクト・タスク実行の両方からの進行状況情報を共通の作業実行ビューに統合する

点に見られる。

作業実行ビューは、作業プロセスの実行に関係する人に関する本発明の概念から得られる複合利益を包含し、それ自体が本発明の概念を表している。プロジェクト・チームに含まれる人は、自分の作業と、協同する他のチーム・メンバの作業を統合ワークスペースで表示し、個々に割り当てられたタスク用のプロジェクト計画ビューと、表示されたタスク内のプロセス実行状態とを表示するワークスペースによってサポートされる。自分のために活動化されたタスクとその実行のために可能になった活動との表示から、次に自分が作業のどの断片を選択するかを選択することができる。

作業プロセス・モデルは、作業プロセスのすべての定義域用のすべての計画詳細データならびに作業プロセスに含まれるタスクと活動のすべての実際のデータ、日付、状態を含む。したがって、作業プロセス・オブジェクトは、通常のデータ・ベース照会機能によって照会可能な、そのオブジェクトによって表されるプロジェクトの履歴に関するすべての計画および実行情報のための包括的なデータ・ベースも表している。

#### 図面の簡単な説明

第1図は、一連のスキーマおよびインスタンスを示し、作業プロセス・オブジェクト（WPO）がステージごとにどのように進化するかを示す図である。

第2図は、建築プロセス・スキーマのグラフ表現を示し、活動タイプと文書タイプを示す図である。

第3図は、第2図のスキーマへのプロセス・インスタンス・マッピングのグラフ表現を示す図である。

第4図は、第3図の3次元表示への偏向を示し、バンドル用の複数のスレッドが目に見えるようになっている図である。

第5図は、第2図、第3図、第4図によるタスク・エンベロープを示す図である。

第6図は、第2図、第3図、第4図、第5図によるタスク・チャートを示す図である。

第7A図～第7C図は、家屋建築プロセス・スキーマ用の作業プロセス・オブジェクト（WPO）のプロセス・スキーマ・セットを示す図である。

第8A図～第8E図は、家屋建築プロセス・インスタンス用の作業プロセス・オブジェクト（WPO）のプロセス・インスタンス・セットを示す図である。

第9A図～第9C図は、家屋建築プロジェクト用の作業プロセス・オブジェクト（WPO）のプロジェクトを示す図である。

第10図は、コンピュータ・システム上で本発明の概念を実施することが可能な実施態様の1つを示す図である。

#### 好ましい実施例の説明

添付図面に関連して、本発明によるプロセスおよびプロジェクト管理のための方法の好ましい一実施例について以下に説明する。

第1図は、一連のスキーマおよびインスタンスを示し、作業プロセス・オブジェクト（WPO）が以下のステージごとにどのように進化するかを示している。

プロセス・スキーマ101

プロジェクト・スキーマ102

プロセス・インスタンス103

プロジェクト・インスタンス104

プロジェクト・インスタンス・グロス（複合タスクを含む）

プロジェクト・インスタンス実行可能（単純タスクを含む）

プロセス・インスタンス実行済み105

プロジェクト・インスタンス実行済み106

使用する参照符号によれば、第1図に示すように、プロセス・スキーマ101から、プロジェクト・スキーマ102が得られる111、プロセス・インスタンス103が得られる112。プロジェクト・インスタンス104が、プロジェク

ト・スキーマ102から得られる113、プロセス・インスタンス103からも得られる114。プロジェクト・インスタンス104は、プロセス実行105の基礎116であり、プロジェクト実行106の基礎である。



各ステージは定義域によって移植されるが、これは上から下へ作成され、概念上は定義域境界によって区切られる。特定の定義域境界は、特定の作業プロセス・ビューがオブジェクトに関して実行された程度を定義し、整合性があり検査済みの属性を備えた定義域内のステージの区分を移植する。ビューの対話を実行すると、通常の場合、定義域境界が上から下へ移動する。このようなビューの一部は、同じ作業プロセス・オブジェクト（WPO）上で同時に機能することができる。通常、記述すべき例外がいくつかあるが、これらの境界は、それぞれのマッピングにより上から下へ順序付けられた位置に存在しなければならない。

定義域は、第1図の順序で上から下へ進化する。各定義域ごとに、作業プロセス・グラフの各活動タイプ要素は定義域固有の層を含み、その層自体が定義域固有の属性を含む。したがって、その活動タイプの層は、その上にプロセス・スキーマ101、活動インスタンス、プロセス・インスタンス103、プロジェクト・インスタンス・グロスおよび実行可能、プロセス実行、プロジェクト実行の各層が置かれる基本層である。場合によっては、一時的および移動可能境界とタスク・エンベロープ情報を管理するために副層を使用する。一部

のケースでは、このような層に対して作業プロセス・オブジェクト（WPO）内のサブオブジェクトが関連付けられる。

「作業プロセス」という用語は、以下の説明では、本発明の概念に基づくプロセス管理とプロジェクト管理の完全統合に使用する。その例は、建築請負業者が所有者の要求に応じて家屋を建築するという建築業界における建築事業のコンテキストから取る。簡潔にするため、サンプルでは単純化したプロセスとプロジェクトを使用するが、本発明の請求項の重要な態様を含むものとする。

選択した業界のうち、家屋および家屋のセットを建築するプロセスについて説明するが、この場合、家屋の建築用に十分定義された標準的な建築プロセス・スキーマ101が適用され、このプロセス・スキーマ101は、標準的な家屋の修正に関する顧客の要求に応じて家屋の具体的な1つのインスタンスまたは複数のインスタンスの建築に対してその建築プロセスを適応する各インスタンスごとにカスタマイズすることが可能である。

標準的な家屋を建築するプロセスの場合、十分に定義したスケジュールおよびカレンダーの制約の範囲内の建築チームおよび建築機器として、資源の適用を管理するプロジェクト・スキーマ102が定義される。プロジェクト・スキーマ102は、通常の条件下で家屋を建築する1つのインスタンスについて想定したデフォルト・データを含む。単一家屋または大規模建築プロジェクトの1組の家屋セットを建築するため

に建築プロセスを適用する各インスタンスごとに、特定の建築プロジェクトに適用された資源の制約に基づいて、家屋を建築する各インスタンスのカスタマイズ要件を反映したプロジェクト・スキーマ102のインスタンスが適用される。

プロセス・スキーマ101、プロセス・インスタンス103、プロジェクト・スキーマ102、プロジェクト・インスタンス104からなるこのセットアップ全体は、これらすべての概念のコンピュータベースの実施態様によって管理され、制御される。関係する各個人、すなわち、プロジェクト内で管理される各チーム・メンバごとに、その人の作業に関する適切なビューを実現する必要がある。このようなビューは、具体的なプロジェクト、ならびにプロジェクト・チームのメンバにより特定の作業専用のワークスペースによるこのようなプロジェクトの同様のすべてのインスタンスのための建築事業に関するすべてのプロセスおよびプロジェクト計画の計画、実行、監視を含む。

このような概念を実施するには、プロジェクトの各インスタンスの実行全体にわたってそれらが計画され記録されるので、すべての計画および実際のデータとともに、スキーマ、そのインスタンス、プロジェクト・スキーマ、そのインスタンスに関するすべてのモデルを含むデータ・ベースとして作業プロセス・オブジェクトを実現することが必要である。以下に示すように、建築事業用の建築資源の最適割当てには、事業に含まれるすべてのプロジェクトにわたってこのような

データを管理することが必要である。というのは、資源管理の最適化は事業全体についてしか達成できないからである。

このデータ・ベースの周囲には、プロジェクト・チーム内のすべてのメンバ用

ならびに事業内のプロジェクト間の監視作業用の個々のワークスペースを実現するために、多くの明確なユーザ・ビューをサポートし、実現する必要がある。

プロセスおよびプロジェクト・スキーマおよびインスタンスの基本概念については、以下に説明する。

まず、プロセス・スキーマ101の概念が定義される。このプロセス・スキーマ101は、プロセスを実行するインスタンスで実行すべき重要な活動タイプと、プロセス内のステップを制御する文書の文書タイプとを定義する。各活動タイプは、プロジェクト内のプロセスを実行するために最終的に必要な人またはプロセッサ・タイプ用のユーザ役割に関連する。プロセス・スキーマ101は、そのスキーマの特定の複数スレッドを何回も並行して実行できることを示し、このようなバンドルの開始活動と終了活動を示す。現在の例では、バンドルという概念は、建築複合体において1組の家屋セットを並行して建築できるが、いくつかの活動は複合体全体に共通するという状況をモデル化するために使用する。この場合も各家屋は、すべての階について骨組み（ROC）が建築され、建築中にも家屋を保護するためにすべての階の上に屋根が建築された後、個々の階ごとに複数活動のバンドルのスレッドを並行して実行できるようなやり方で建築可能な、複

数の階を備えている可能性がある。

第2図は、建築プロセス・スキーマ101のグラフ表現を示し、活動タイプと文書タイプを示している。個々の活動タイプ、文書タイプ、ユーザ役割には、家屋を建築するプロセスにおけるこれらの項目の意味を直観的に理解するために、ここでは自明と見なされる名前が与えられる。

以下の省略形は、次のように自明の長い名前に対応する。

CAR	214	大工の作業
EMO	219	外部モルタルの塗布
FLO	203	床の青写真
HDE	201	住宅団地の青写真
HEA	216	暖房装置の取付け
HOU	2●	家屋の青写真 ●

IMO	217	内部モルタルの塗布
PAI	218	内部の塗装
ROC	212	骨組み
ROO	213	屋根構え
SAN	215	衛生設備の取付け
UPE	204	使用許可
USE	220	使用許可検査
UTI	211	ユーティリティおよび基礎工事

第2図で使用する記号は活動タイプと文書タイプを示し、同一記号であるがインスタンス名が付いているものは、後でプロセス・インスタンスに使用する。すなわち、HDE20

1、HOU202、FLO203、UPE204は「文書」を表し、UTI211、ROC212、ROO213、CAR214、SAN215、HEA216、IMO217、PAI218、EMO219、USE220は「活動」タイプを表す。SPLIT HOUSE231とSPLIT FLOOR232は「バンドル分割」を表し、JOIN FLOOR235とJOIN HOUSE236は「バンドル結合」を表し、SPLIT INSTALL233は「分割」を表し、JOIN INSTALL234は「結合」を表す。建築資材の流れは、本発明の概念に貢献しないと思われるので、簡潔にするため、これについてはモデル化していない。

プロセス・スキーマ101を実行する前に、まず、概念上、そのスキーマをコピーすることにより、まずスキーマのインスタンスであるプロセス・インスタンス103を生成する必要がある。このコピーは、そのプロセスの本質を大幅に変更しないような何らかの修正を標準的なプロセスに加えることによってカスタマイズすることができる。このようなカスタマイズは、標準的なプロセス内で提供された様々な顧客の意見を反映することができる。この場合も、このようなカスタマイズは本発明の考え方に貢献しないので、簡潔にするため、これについてはこれ以上説明しない。

次に、プロセスの●スタンスで実行すべき具体的な活動●プロセスのこのイ

インスタンスを充填しなければならない。建築する家屋の軒数と各家屋の階数に応じて、特定のインスタンスで実行しなければならないバンドルの並列スレッドの

数を決定しなければならない。この場合、プロセス構造が変わらなくても、具体的な活動はカスタマイズ要件を反映しているはずである。定義済み活動のそれぞれを定義済みの役割に割り当てなければならない。一般に割り当てはプロジェクト・スキーマ102から引き継がれるが、所与のプロセス・インスタンス103用の特定のインスタンスに応じて変更可能である。また、文書タイプの参照も、特定の建築プロジェクトの文書を参照して、具体的なものにしなければならない。

ここに想定する例はカスタマイズによりプロセス構造を変更することはないので、第3図に示すプロセス・インスタンス103のグラフ表現は、垂直に表示すると、第2図のスキーマに直接マッピングされる。

第3図のUTI活動311は第2図のUTI活動タイプ211に対応する。第3図のROC. 1 312. 1およびROC. 2 312. 2活動は第2図のROC活動タイプ212に対応し、この場合、「. 1」という表記は「第1の家屋」を意味し、「. 2」という表記は「第2の家屋」を意味する。同様に、第3図の4つのCAR活動314. 1. 1、314. 1. 2、314. 2. 1、314. 2. 2は第2図のCAR活動タイプ214に対応し、この場合、「. 1. 1」当表記は「第1の家屋の1階」を意味し、「. 1. 2」という表記は「第1の家屋の2階」を意味し、「. 2. 1」という表記は「第2の家屋の1階」を意味し、「. 2. 2」という表記は「第2の家屋の2階」を意味する。簡潔にするため、活動

と文書はSPLIT INSTALL項目333のみに完全に限定される。

第3図を3次元表示に偏向すると、第4図に示すようにバンドル用の複数のスレッドが目に見えるようになる。第2図では、文書と活動の名前は家屋の番号と家屋内の階数で修飾され、2階建ての2軒の家屋を想定している。一部のスポットに限定して、このように修飾された名前をすべて示されている。たとえば、ROC. 1 412. ●は家屋1での「骨組み」活動であり、●CAR. 2. 1 4

14. 2. 1は家屋2の1階での「大工の作業」活動である。第4図のプロセス・インスタンス103の偏向表現では、活動と文書の名前がほとんど省略されている。

実際のプロジェクト環境のプロセス・インスタンス103は、以下の説明では簡略にするためにプロセス103とも呼ぶが、プロジェクトに使用可能な実際の資源のコンテキストで実行する。このようなプロジェクト用にデフォルト・スキーマであるプロジェクト・スキーマ102を用意すると便利である。これは、それから実際のプロジェクト用の計画を得ることができる113、共通のプロジェクト計画スケルトンとして使用する。現在の例のプロジェクト・インスタンス104は、以下の説明では簡略にするためにプロジェクト104とも呼ぶが、プロセス103から直接得られる114。プロジェクト104は、まとめてスケジューリングし、割当て可能な作業単位であるタスクに取り組む同じ人またはチームに割り当てるべきタスクにする活動のグループ化を特定する。

以下の説明では、「資源」という用語は、同一タスクに割り当てられる人またはチームである、固有の人的資源を示すために使用する。一例として、コンクリート職人のチームは、各個人ごとにさらに正規の計画を行わずにタスクに取り組む。このようなチームは、資源管理に関して確立した概念である。現在の例では、プロセス103に関するプロジェクト104は、ともに第5図でタスクH1 F1 SAN&HEA5156. 1. 1（この場合、H1 F1という表記は「家屋1の1階」を意味する）または第2の家屋用の作業でタスクH2 SAN&HEA5156. 2が示すように、1つのスレッドで衛生設備の取付けと暖房装置の取付けの両方に割り当てられる2つの活動を役割導入時に同一チームに割り当てる唯一のケースを示している。同様に、複数の活動を四角形内に囲むことができるが、その四角形は名前付きタスクを表し、そのタスクに集約した活動を囲むものである。

第5図に示すように、この単純な例でも、複数の活動からなるバンドルに関してより複雑な状況が存在する。第5図ではタスク・エンベロープが見えるようになっている。たとえば●家屋1の1階の内部モルタルの塗布●内部の塗装の場合

はH1 F1 IMO&PAI5178. 1. 1であり、家屋2の1階と2階の内部モルタルと外部モルタルの塗布の場合はH2 IMO&EMO517. 2である。プロジェクト・スキーマ102は、プロセス・スキーマ101を実行するためのデフォルト値を含むので、一般的な場合では建築すべ

き家屋の数と各家屋の階数を把握することができない。この拡張は、特定のプロジェクトに関するプロセス・インスタンス103の定義によって既知のものになる。バンドルを複数のスレッドに拡張することは、プロセス・インスタンス103の定義で行われる。以下に示すように、スレッドの拡張と、この拡張へのタスクの割当ては、いくつかの様々なオプションを提供する。当然のことながら、実際のプロジェクトでは、建築活動のいくつかがより小規模の補助活動に分解され、タスク割当ての対象となる。しかし、この場合も、これ以上詳細にしても本発明の基本概念に貢献しないものと思われる。

プロジェクト・スキーマ102は、1つのタスクに集約される活動を含むプロセス・スキーマ101に関して、この例で四角形で図形表現されている閉鎖境界を定義することによって、プロセス・スキーマ101に対してマッピングされる。プロセス・インスタンス103に対するプロジェクト・インスタンス104のマッピングの場合も、同じ境界概念が関与する。



実際のプロジェクトの場合、そのプロジェクトで使用可能な実際の資源は、活動を実行し、その可用性と完全利用に応じた割振りを管理するための様々な役割に割り振る必要がある。これは、プロジェクト・スキーマ102をインスタンス化し、その結果、プロジェクト・インスタンス104を作成するか、またはプロセス・インスタンス103からプロジェクト・インスタンス104にマッピングすることによって得

ることができる。ここでは、実行のインスタンス用に移植され、選択されたプロセスは、プロジェクトの進行中に修正されないものと想定する。プロジェクト実行中のプロセスの動的改良がどのように達成できるかについては、後で示すものとする。

プロジェクト全体にわたって、プロジェクト計画、すなわち、その関連データをすべて備えたプロジェクト・グラフは動的に変更される。このような変更の規則については、これまでに述べた詳細に必要なデータ・ベースの実現を紹介した後で説明する。データ・ベース構造を拡張する追加の詳細については、後で説明する。関連のプロセス・スキーマ101、プロセス・インスタンス103、プロジェクト・スキーマ102、プロジェクト・インスタンス104に関する事業のすべての情報、すなわち、全運転情報は、概念上、この例では建築事業のすべての建築プロジェクトに関し、プロジェクト履歴、現行状況、今後の計画に関するすべての情報を含む単一作業プロセス・オブジェクト（WPO）として表示することができる。

このような作業プロセス・オブジェクト（WPO）は、実際には、すべての履歴情報を含む、建築事業の全運転データ・ベースを表す。実用のため、この情報の一部は老化し、ある程度の時間が経つと無関係のものになる。実際の状況では、データ・ベースのオーバフローを回避するために、この情報の一部を時々削除する必要がある。しかし、このような考慮

事項は本発明の概念にとって重要なものではないので、データ・ベースで管理される情報についてのみ説明するが、このような情報がデータ・ベースから削除される、すなわち、作業プロセス・オブジェクト（WPO）から消失する時点については説明しない。

第2図、第3図、第4図にすでに示したように、プロセス・スキーマ101、プロセス・インスタンス103、オーバーレーしたプロジェクトの基本概念は、有効グラフとして表示することができる。この方向は、グラフを解釈したときに活動の実行順序を示す。実行の方向は、実行済み活動から非実行活動へ向いている。これらのポイントは、再実行または実行再開のためにすでに実行した活動を指し示すことができるという点で逆方向を指し示すこともできる。したがって、実際には、グラフはループを含むことができる。第3図の例は、第4図にも示すように順番に同じコミッションにより行われる、2軒の家屋の受入れ検査のためのループUSE 32  示している。 



たとえば、関係データ・ベースでこのようなグラフを実現すると、その結果、「活動」と「有効コネクタ」という項目タイプが得られ、これらが関係を形成し、さらに様々な属性を保持することができるはずである。各項目タイプごとに、関連属性の実際の値に関連付けられた明確な項目インスタンスがデータ・ベースで管理される。

第7図および第8図は、第2図、第3図、第4図用の作業

プロセス・オブジェクト（WPO）の内容の一例を示し、2階を有する2軒の家屋を建築する場合にこのプロセスを使用するための作業プロセス・オブジェクトを示している。この作業プロセス・オブジェクト（WPO）モデルのタスク用の境界は、第3図からのプロセスのサブセクションとして後で第5図に示すように、それに基づいて特定の活動が実行されるタスクを識別する、特定の活動の属性としても表される。

作業プロセス・オブジェクト（WPO）データ・ベースは、プロジェクト内の実行の進行状況に関する情報を含むが、作業プロセス・オブジェクト（WPO）をどのように構築し、それをどのように修正できるかを制御する何組かの規則によって増強する必要がある。このような規則は、プロジェクトを計画し実行する際にユーザが実行できる様々なアクションを制御し、様々なユーザ機能をサポートするために実現した様々なビューによって最もよく記述することができる。

このようなビューとユーザ機能は、必ず作業の進行状況を定義し、計画し、制御するが、ほとんどの場合、現在述べている概念における人間または機械による実際の手動作業に影響することはない。言い換えれば、ここで述べるプロセスおよびプロジェクト管理システムは、煉瓦職人がどのように煉瓦を積むか、またはどのように煉瓦積み機械を使用するかを制御するものではなく、むしろ、全体的なプロジェクト計画により、いくつの煉瓦を何時間で積まなければならないかを制御するものである。当然のことながら、この職人の作業は

第6図に示すタスク・チャートで目に見えるようになっている建築計画などの管理文書によって管理されるが、第6図に示すタスク・チャートで目に見えるよう



になっている建築計画を解釈し、自分の専門技能により自分の作業を実行することはその職人の責任である。プロセスは、計画に応じて作業が実行されたことを検査する諸要素を含み、最終的な使い方に要約されたこの例では、完成した家屋に対してUPE304. 1および304. 2を行うことができる。当然のことながら、実際のプロジェクトでは、再作業により後で行うエラー—訂正が蓄積するのを回避するために十分早い時期に、第6図に示すタスク・チャートで目に見えるようになっている建築計画からの逸脱を検出するために、もっと多くの検査がプロセスに組み込まれる。

プロセスおよびプロジェクト管理の概念は、プロジェクト内の所定のレベルの細分性までは妥当であるが、単一煉瓦を積み重ねるという最低レベルまで下がることはないことをこの例から認識することは、重要な態様の1つである。細分性の深さは、制御の細密細分性による利点に対する計画および制御のオーバーヘッドを評価する、実際の考慮事項によって決まる。まったく同じ考慮事項は、建築業界という現在の例から、ユーティリティ、政府機関、造船を経て、ソフトウェア開発まで、同じ手法を使用するすべての同様のビジネス分野にも適用される。

プロセスおよびプロジェクトを開発するためのユーザ・ビ

ューについては、次のように説明する。

これまでに紹介した様々な概念とこの例の後半に続く改良は、実際のプロジェクトのインスタンスでこれらを実行するすべての詳細により、建築プロジェクト用のデフォルト・スキーマである建築プロセス用のスキーマの一般概念をまず開発する際に、様々な専門的な役割の人間の作業を含む。それぞれの役割は、全体的な作業プロセス・オブジェクト（WPO）のビュー関連部分を作成し修正することができ、特定の役割でユーザが実行できるアクションの指導と制約の両方を行うような様々な規則を含む、複数组の対話に関連付けられている。

このような規則の大部分については、完了したオブジェクトだけがその規則に従わなければならない、指定中は一部の規則は実施しなくてもよい場合もある。各オブジェクトごとに、完了状態が定義され、すべての規則が実施される。編集中のオブジェクトは、中状況に戻ることができる。ビュー動的開発では、完

了境界により、いまだに作業中の下部区分から完了状態になっている上部区分が定義される。役割には、このような役割の特定の機能を意味的に示す役割名が与えられる。役割と役割名の概念は、ここではIBM FlowMark製品と同じ意味で使用する。

プロセス・スキーマ設計ビューは、第2図に示す特定の建築プロセス・スキーマ101などのスキーマを設計し編集するための目的に対応するものである。このビューに関連する

ウィンドウでは、この役割のユーザが、先行活動と後続活動との関係が必ず十分に定義されている有効グラフを形成するようなやり方で活動タイプ、制御コネクタ、分割、結合を配置することができる。さらに、ユーザは、様々な属性を活動およびコネクタに関連付けることができる。第7図および第8図は、ビュー・ユーザが値を指定できる、許可された複数組の属性を示している。

グラフ内のコネクタは、第2図の例に示すように、特定のノードで分割および結合することができる。この分割／結合の概念は、本発明の概念にはあまり貢献しないので、現在の例ではあまり使用しない。分割に関連する態様の1つは、「m個の出力から1個選択」（分割記号中に「1」という記号で表す）として前述したように1つの分割からの出口が互いに排他的であるか、1つの分割から選択した複数の代替案（「#」記号で表す）が可能であるか、または「すべての出力の選択」（「A」記号で表す）のように処理の実行後に分割からのすべての出口が実行されるかという特性である。同じ概念は結合にも適用される。このような分割／結合の概念は、分割後の活動に関連するようにタスクを定義するときに適切なものになる。

ここに紹介した新規の概念は、先行活動が終了する前に後続活動の開始を可能にする「先行作業」タイプのコネクタと、通常の制御コネクタという2つの並列制御コネクタによって活動を接続することにより、活動の実行をオーバーラップさせ

るための手段である。先行作業コネクタは、通常の制御コネクタに加えて使用する

る場合のみ可能であり、作業プロセス・オブジェクト（WPO）内にコネクタ用の先行作業属性として表される。先行作業タイプのコネクタは、グラフ表現に特殊記号によって表すことができる。詳しくなりすぎるのを回避するため、添付図面ではこのような区別を付けていない。

オーバーラップ・コネクタによって開始された活動は、先行活動が終了したという信号を通常のコネクタが送信した場合のみ終了することができる。言い換えれば、先行活動は、先行作業用に1つ、正規の順序付けに1つという2つの信号を送信するようにグラフに設計しなければならない。先行作業の概念は、タスク・スケジューリングにも関連する。第2図のグラフは、第7図の表に示すように、制御コネクタ、分割、結合によって関連付けられる活動タイプ項目によって作業プロセス・オブジェクト（WPO）に表される。


プロセス・スキーマ101のグラフを作成し編集する場合、いくつかの規則が適用される。

非コピー規則では、具体的な実行のためにスキーマがまだインスタンス化されていない間、すなわち、作業プロセス・オブジェクト（WPO）にスキーマのコピーが一切に作られていない間だけ、プロセス・スキーマ101のオブジェクトを修正できなければならない。

分割／結合整合性規則では、グラフ内の対応する分割と結合が同一かまたは論理的に競合しない特性「1」、「#」、

「A」を備えていなければならない。

接続性規則では、グラフの「開始」または「終了」活動として示された活動を除き、少なくとも1つの入力制御コネクタと1つの出力制御コネクタによってすべての活動を接続しなければならない。

プロセス・インスタンス設計ビューは、プロセス・スキーマ101が完了状態で存在することに基づいている。このスキーマにより、元のスキーマを実行可能にするようなすべての詳細名と属性を定義することができる。これは、これらの属性ならびに特定の属性値に参照される実際の活動、コネクタ、オブジェクトを含む。このビューは、体的なデータ、名前、値によるスキーマの移植を実行す

る。まず、対話は、第8図に示すように第2図から作業プロセス・オブジェクト（WPO）内のインスタンスヘスキーマをコピーし、そのインスタンスに関して作成したコピーを指し示す同じグラフ表現を使用し、そのインスタンスを移植する。

具体的には、並列スレッドのインスタンスをいくつ実行すべきかをバンドルについて定義しなければならない。この定義のため、第3図に示すようにスキーマのコピーが3次元グラフに拡張され、作業プロセス・オブジェクト（WPO）の内容については属性と値を第8図に示す。活動または制御コネクタをクリックすると、このビュー用の補助対話がオープンするが、これはデータ、属性、値の入力をサポートし、第3図による2次元表現から自動的に作成されるバンドル内の

並列スレッドの数に関するものである。

プロジェクト・スキーマ・ビューでは、そのプロジェクト（インスタンス）で使用する、複数のプロジェクト・データを定義することができる。この例を簡潔にするため、プロジェクト（インスタンス）ビューについて説明するが、プロジェクト・スキーマ定義の詳細までは説明しない。プロジェクト・スキーマは、プロセス・スキーマに関して定義され、個別の作業プロセス・オブジェクト（WPO）のサブオブジェクトによって表される。

プロジェクト・タスク定義ビュー： プロジェクト・ビューは、「完了」状況のプロセス・インスタンス103に関して定義される。第5図は、タスク境界によって1つまたは複数の活動をエンベロープ化し、第9図に示すように個別のタスク・オブジェクトを作業プロセス・オブジェクト（WPO）に設定することによって、プロセス・ビューに関して1組のタスクをどのように定義できるかを示している。たとえば、活動ROCをエンベロープ化することにより、タスクH1

ROC512.1が定義される。この場合、H1 ROCという表記は第1の家屋の骨組みを意味する。同様に、第2の家屋について活動SANおよびHEAをエンベロープ化することにより、タスク・セットH2 SAN&HEA5156.2が定義される。これらのタスクは、この例のH1 ROC612.1およびH2 SAN&H2 6156.2として、第6図のタスクチャートにも示

されている。

以下の説明では、すべての活動へのタスクの完全割当てのいくつかの態様についてのみ、説明する。プロセス・インスタンス103に関してタスクを定義する場合、1組の規則が適用される。

閉鎖エンベロープでは、例において四角形として描いたようにすべてのタスク・エンベロープは閉鎖線でなければならない。

非交差では、タスク・エンベロープは互いに横切ったり交差してはならない。

非結合タスクは、2つまたはそれ以上の完全なエンベロープが1つのタスクに属するように定義することができる。第6図に示すように、例では、H2 IMO & EMO 5 1 7. 2とH2 IMO & EMO 5 1 9. 2という2つのエンベロープがある。

ネストでは、基本タスクからなる複合タスク内にタスクをネストすることができる。しかし、この特徴は例には示されていない。というのは、このネストは本発明の概念に貢献しないが、特定の追加規則を必要とするからである。詳しくは、以下の動的ビューで説明する。

固有資源では、特定の役割の人またはすべてが同じ役割を果たす人からなるチームという単一資源にタスクを割り当てなければならない。タスクへの活動のカプセル化は、同じ役割に関連する活動だけをカプセル化することを主張することにより、この規則に従わなければならない。物理的人物への

割当ては、ここで行うか、または第9図の作業プロセス・オブジェクトに示すタスク割当てビューで後で行うことができる。

タスク・プラグマティズムでは、タスクの定義は無差別に行うのではなく、意味のある細分性の作業管理に応じて行わなければならない。作業単位は、事業の管理に値する量の労力を必要とする。前述のように、単一煉瓦の積み上げは管理する必要がないが、むしろ、人時、人日、人週単位で測定された作業単位を作業プロセス・オブジェクト（WPO）内にタスク用の労力属性で示す必要がある。

プロジェクト・タスク定義ビューでは、個々のタスクが、定義済みの資源想定

に基づいてタスクを実行するための予測労力および予測経過時間により、そのタスクを実行する役割に関連し、管理すべき作業単位として設定されている。プロジェクト・タスク・スケジューリング・ビューでは、これらのタスクが、最終完了事象まで家屋を建築するプロジェクトに必要なスケジュールとその可用性に応じて、役割を果たす人に割り当てられる。資源管理およびスケジューリングのアクションは、現在のプロジェクト管理ツールでは最新技術である。第6図は、このビューでは一般的な対話タイプを示している。バーは様々なタスクを表し、プロジェクト計画プログラムは、これらのタスクを再配置し、正しい数の人間を割り当てて、プロジェクトの目標に応じて最善の結果を達成することができる。しかし、本発明の概念を反映して、これら

の機能に関する特定の制約を紹介する必要がある。

プロジェクトとプロセスとの整合性では、タスクが時間目盛上に配置されている場合、タスク内に含まれる活動の順序に違反しない程度に限り、タスクをその順序で再配置できなければならない。言い換えれば、プロセスに定義されている論理活動順序は、これらの活動を実行するタスクの順序によって違反してはならない。

タスク・オーバラップは、2つのタスクが時間的にオーバラップ可能な複数の活動を含む場合に可能になる。というのは、これらはプロセス・グラフの2つの並列分岐に入っているからである。

タスク先行作業は、2つのタスクが先行作業コネクタによって接続された活動と先行／後続関係になるような状況を定義するものである。この場合、タスクはスケジュール計画でオーバラップすることができる。というのは、両方のタスクは、含まれる活動と同じく、同時に活動状態になりうるからである。上記の例では、これは、先行作業コネクタ属性によって許可され、対応するタスク定義によって利用されるH1. ROCとH1. ROOというタスクのケースである。

プロジェクト内の活動を実行するためのユーザ・ビュー： これまでに定義したビューは、そのスキーマおよびインスタンスでのプロセスおよびプロジェクトの開発を含むものである。プロジェクトが定義され開始されると、様々な実行ビ

第6図に示すビューの読取り専用の変形態様として実現した作業棒グラフは、建築プロジェクトで●●●している1組のチーム用のグローブ●●●ビューまたはカ



ストマイズ・サブセット・ビューを示す。タスク割当てではカレンダーを基礎とするバー・ダイアグラムとして表示され、各グループはその長さが作業の期間を表すバーとして自分の作業割当てを確認し、このダイアグラムにより、特定のグループに関係する他のグループの作業や、タスクがどのようにオーバーラップするかに関するビューも可能になる。

活動順序リストは、個人またはチーム・リーダー向けに、資源、スケジュール、必要な完了情報に関する十分な詳細データとともに、その要求された実行順序で実行すべき活動のリストを表示するものである。

ビュー定義完了および実行の動的進行： 上記の説明では、インスタンスをコピーするには、その前にプロセス・スキーマ101を完了しなければならないことだけが必要であった。互いにマッピングされた他のすべてのビューについては、オ

ープンのままになっているマッピング用の規則をここに示す。上記で紹介したビューの一部は、残りのビューに関して定義されるものである。実際のプロジェクトでは、基本ビューの一部分がまだ指定中である間に、このようなビューの定義を進行することができる。また、後半部分でまだ定義と指定が続行している間に、個々のビューはすでにその前面部分（有効グラフの観点での前面部分）で実行を開始することができる。たとえば、このような状況进行处理するために1組の規則が用意される。

完了境界は、プロセス・インスタンス103およびプロジェクト・インスタンス104ビュー用の定義の境界であり、完了境界の上で上記の規則により定義が完了するが、グラフの方向が上から下であると想定すると、境界の下部分はまだ定義状態にある、すなわち、不完全である可能性がある。完了状況は、作業プロセス・オブジェクト（WPO）活動オブジェクトに示され、カラーなどで図形表示できるはずであるが、これは現在の黒白の図で提示するには不便である。

ループ完全性では、完全性境界が完了境界の上下の各部分にループを切断してはならない。

プロジェクト定義は、それに関してプロジェクトが定義されるプロセス・

インスタンス103の定義境界を追い越すことができない。すなわち、プロジェクト定義境界は、それが定義されるプロセス・インスタンス103の完了境界の上に必ず存在しなければならない。プロジェクト割振り境界

は、すべての複合タスクが基本タスクに分解され、実行資源がタスクに割り振られている区分を定義する。

もう1組の規則は、プロジェクトの実行境界に関連する。プロジェクトが実行中の状態になっている場合、境界は、どのタスクが開始され、実行中の可能性があるのかを定義する。

プロジェクト進行は、プロジェクト割振り境界を越えて進むことができない。その境界自体は、プロセス・インスタンス103定義完了境界を越えて進むことができない。これは、プロセス・インスタンス103とプロジェクトを計画している人がその責任を持たなければならず、不完全な計画によって実際の作業実行が妨げられないペースで計画が進行しなければならないことを意味する。計画が不完全であると、プロジェクト内に未割当ての資源や空き要員が発生する恐れがある。

このような動的境界は、これらのビューの実現をある程度複雑にするが、プロジェクト開始時の静的定義および資源割振りが達成不能であり、現実的ではないような、十分なサイズや期間のプロジェクトにおける現実的な要件である。上記の様々な境界は、作業プロセス・オブジェクト（WPO）内の活動およびタスク・オブジェクト用の境界の上下にある状況を識別する状況属性によって達成される。1つのビューを表す各対話に関するこの情報から、有効グラフで関連の実際の境界を見つけ、表すことができる。

バンドルに関するタスクは、個別に論じた特殊な状況を表

す。例では、このような状況は第3図のプロセス・インスタンス103で発生し、各階を建築するために、1つのバンドル内の複数の活動からなる並列スレッドが実行されるが、これは各階ごとに同じであるが、各階の作業用のタスクでの実行のために個別に計画することができる。このような状況は第5図に概略を示す

垂直エンベロープ化では、活動がバンドル内の複数スレッドにわたって計画されず、むしろ、複数タスクからなる1つのスレッド内にのみ集約される。このオプションは、関係するチームのサイズが複数階で同時に作業するのに十分な大きさではなく、むしろ、1階ずつ順に終了しなければならない場合に選択することができる。この例では、家屋1を建築するためのいくつかのタスクに関するこのような状況を示している。

この作業プロセス・オブジェクト（WPO）１００１は、その事業のプロジェクト用のすべての静的定義と動的実行値を含む。中央演算処理装置内のワークスペース・ネットワーク・サーバ１００２は、各個人ユーザがシステム１０３０内の自分の役割のニール（ニール）に応じて調製したワークスペースによってサポートされる。

ように、このシステム1030のユーザの個々のワークステーション1021、1022、1023、1024、1025、1026への通信インタフェース1005を提供する。

企業用中央コンピュータ1004内に実現されたデータ・ベース・ビューア1003は、ワークステーション上の自分のワークスペース上に各ユーザが必要とする個別設定ビューを提供する。第10図は、これらのビューが全作業プロセス

・オブジェクト(WPO)1001からどのように分離され、それらが個々のワークスペース1021、1022、1023、1024、1025、1026上にどのように表示されるかを概略で示している。

この例では、不要な詳細を省略し、本発明に関連する態様を強調するように、本発明の重要概念を提示しようとしている。追加の本発明の概念を紹介する必要もなく、完全な現実のプロジェクトに必要なもののために上記の詳細を拡張することは、専門家にとって容易であるはずである。

【図 1】

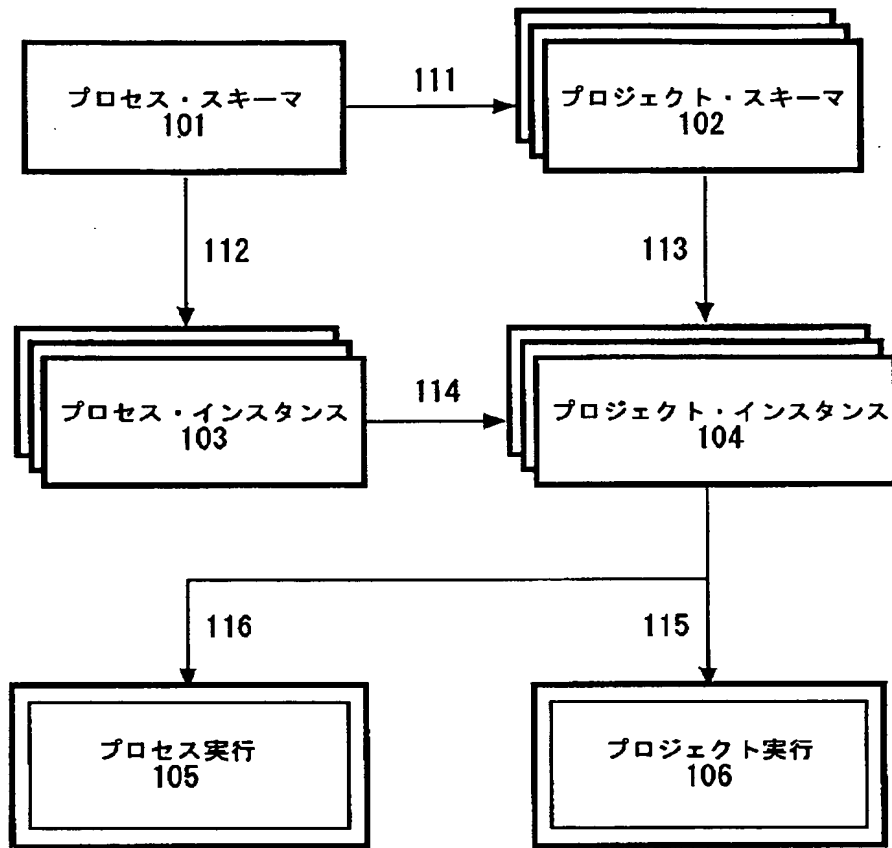


FIG. 1

【図2】

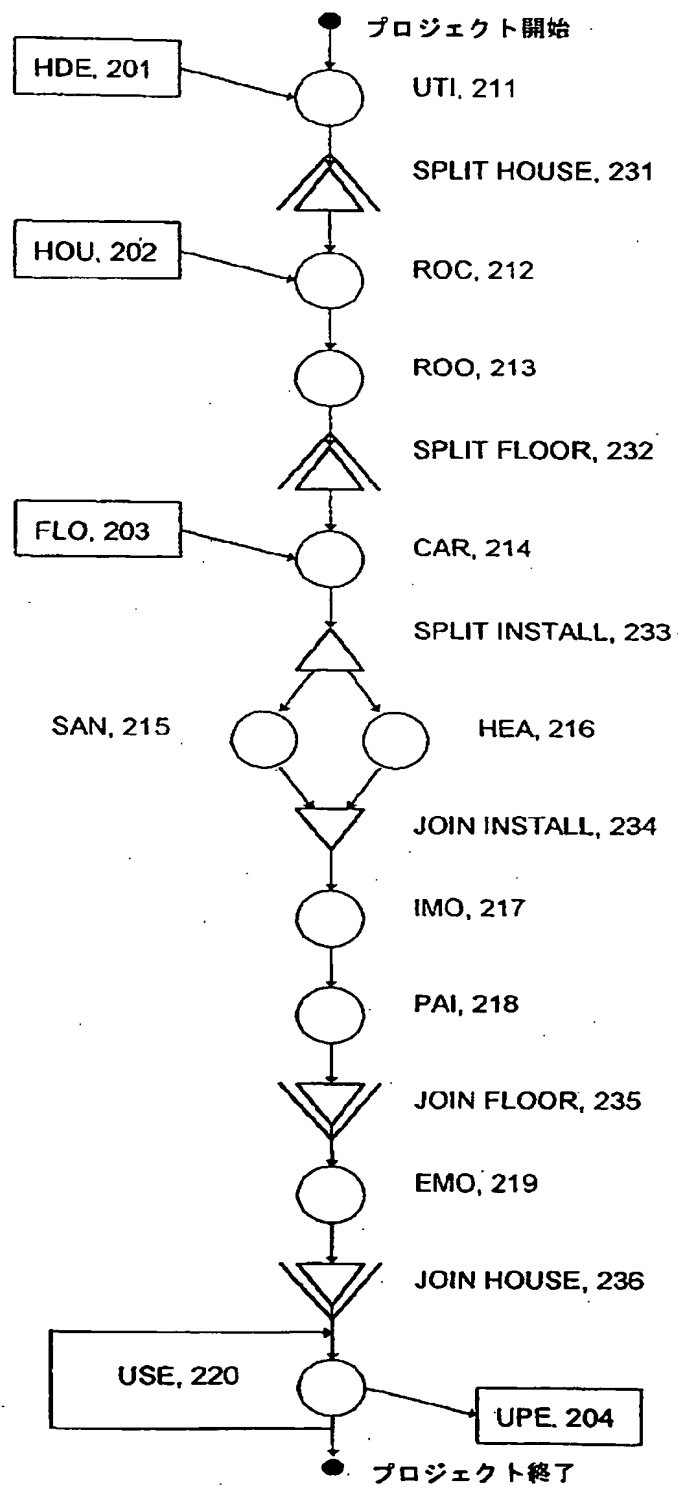


FIG. 2

【図3】

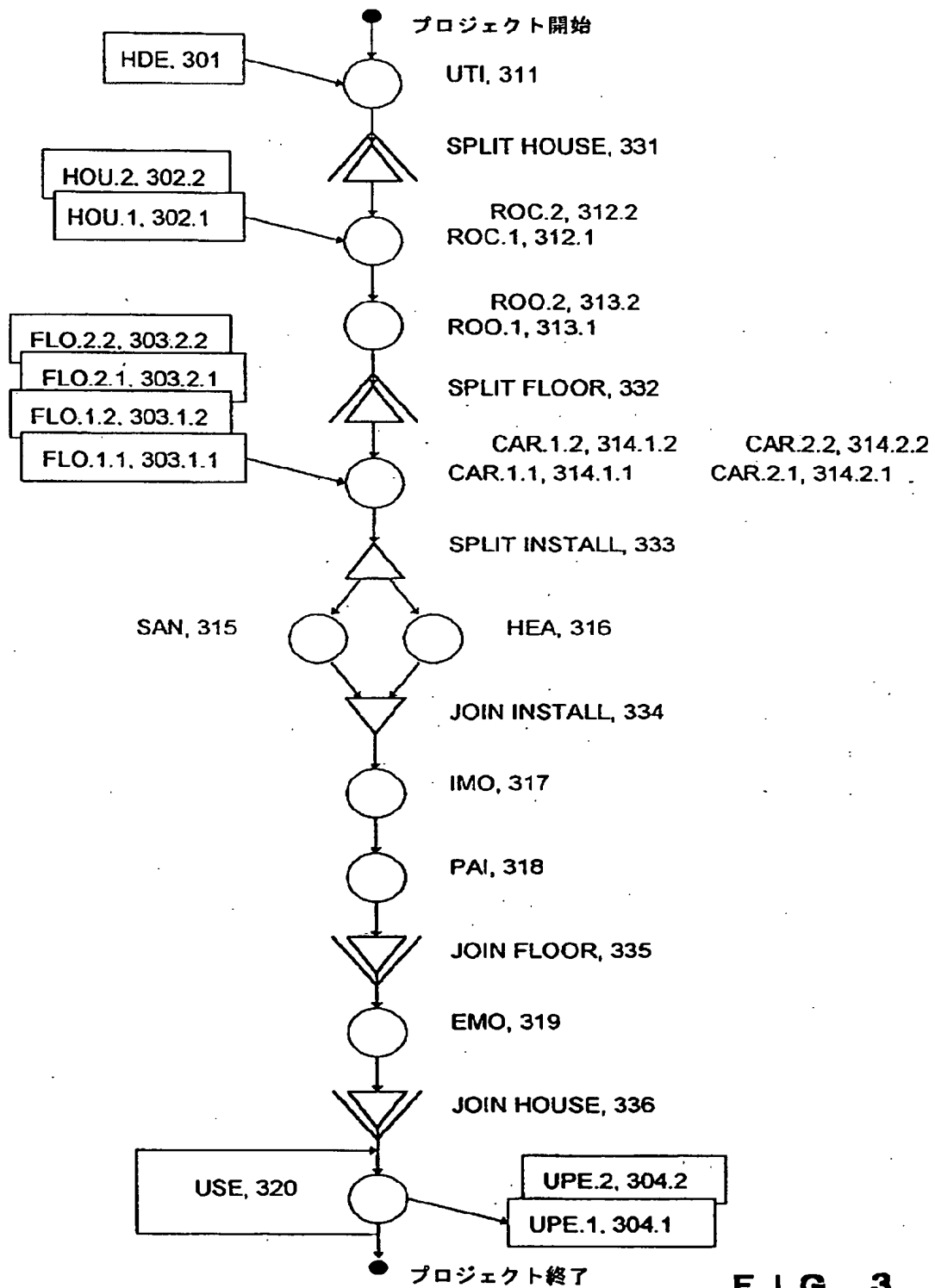


FIG. 3

【図4】

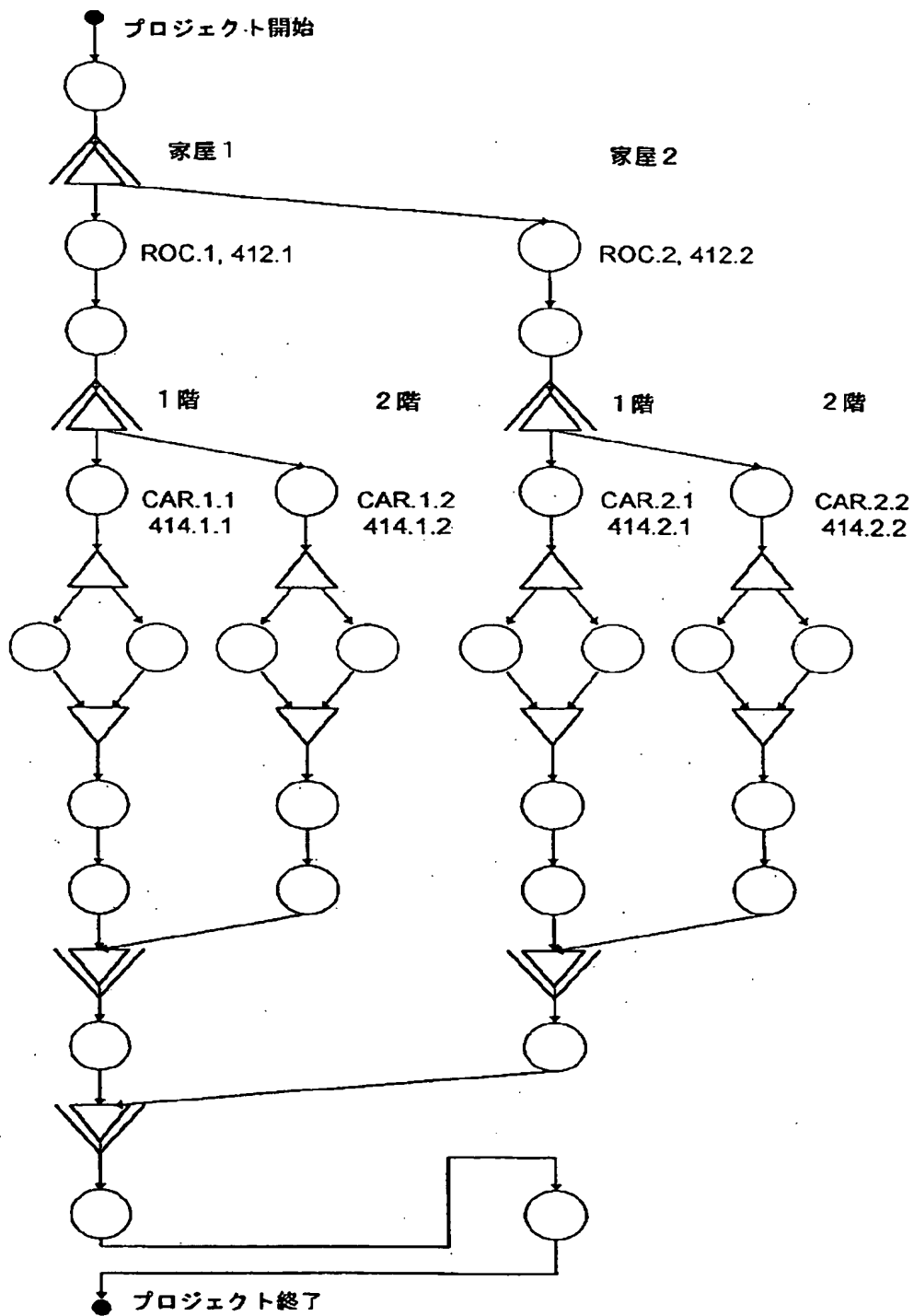


FIG. 4



【図5】

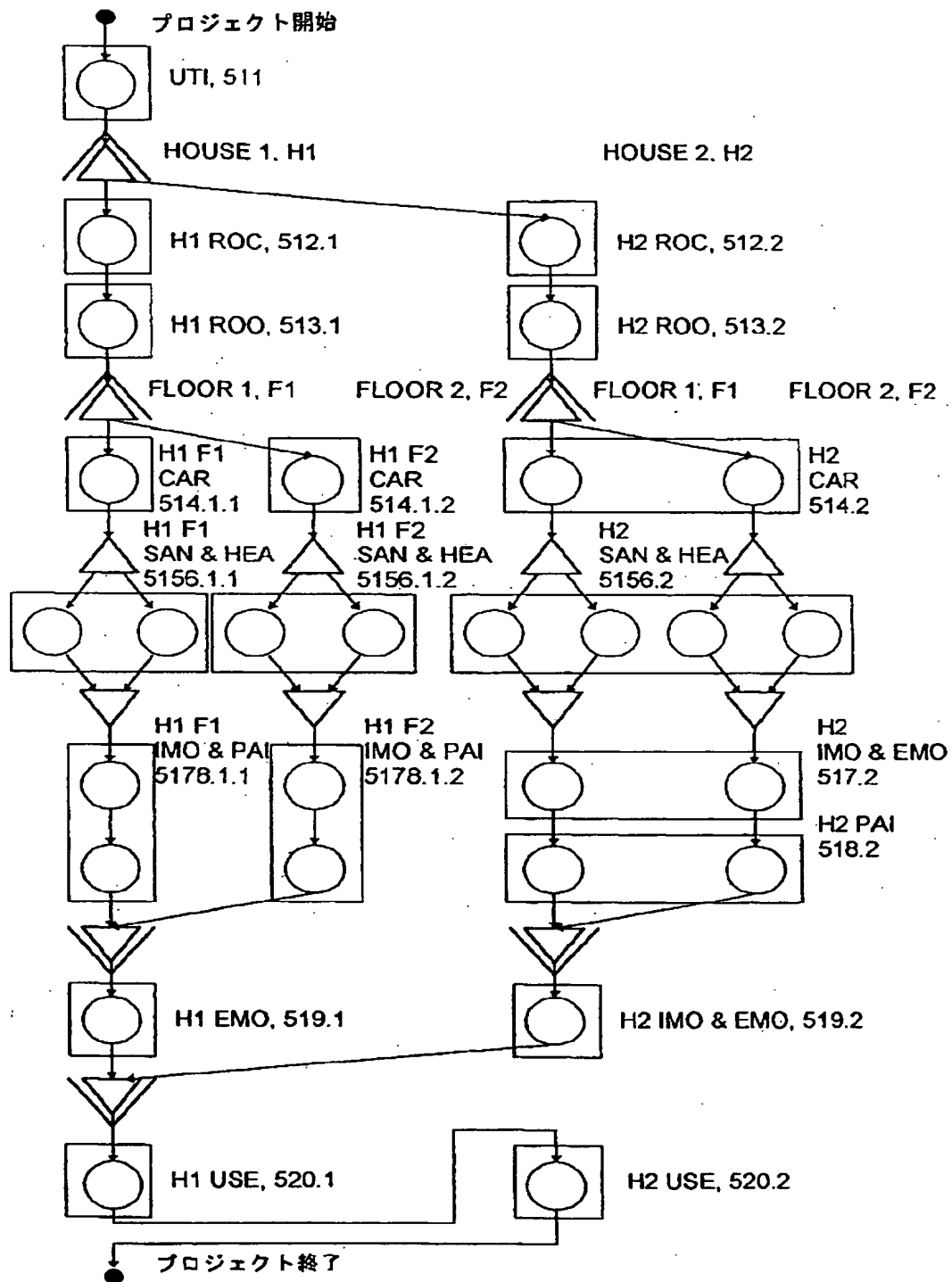


FIG. 5

【図6】

プロジェクト開始

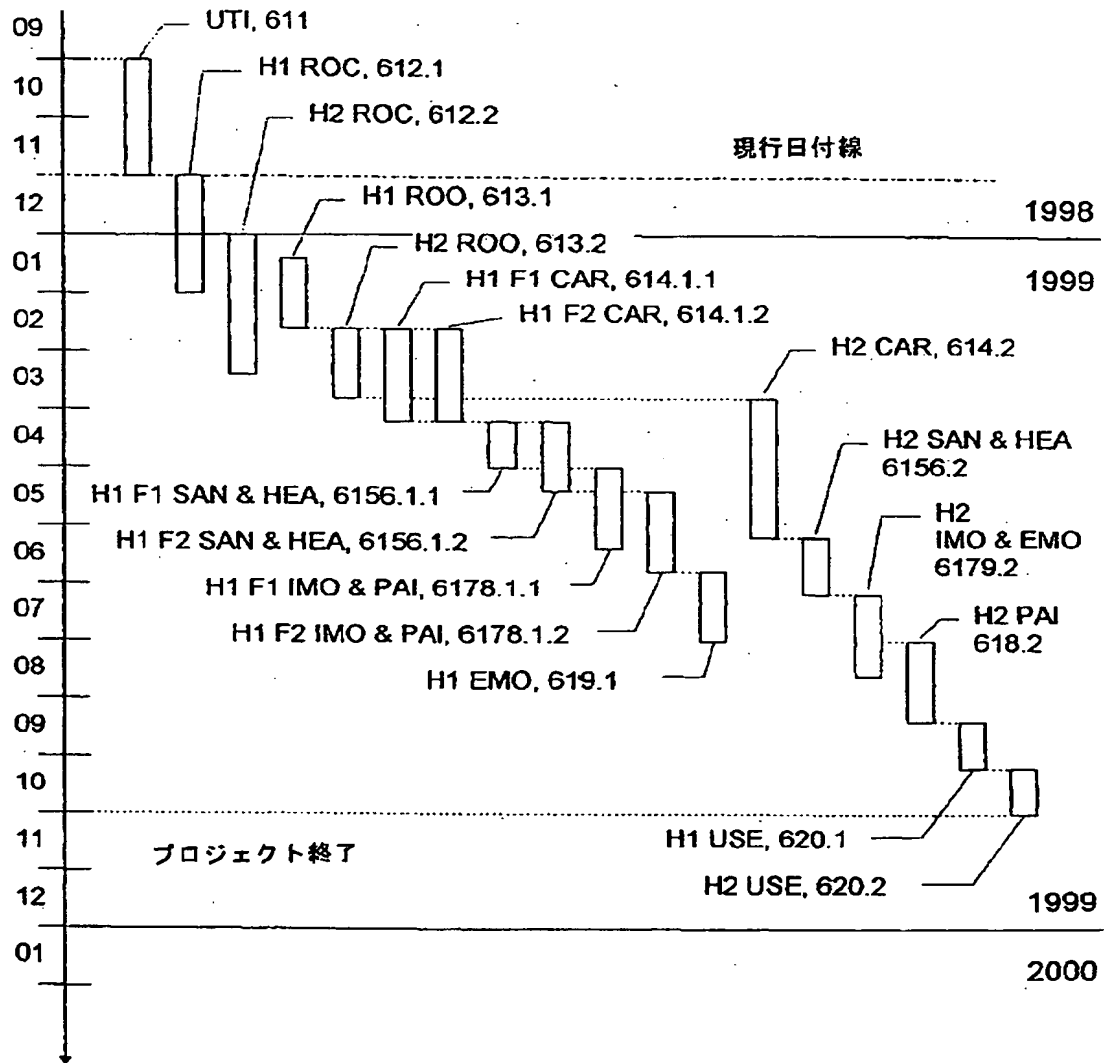


FIG. 6

【図7】

プロセス・スキーマ・セット	
プロセス・スキーマ名	
家屋建築プロセス	
活動タイプ・セット	
活動タイプ名	
活動属性	
活動属性値	
U T I	プロジェクト開始からの 入力コネクタを有する S P L I T H O U S E への 出力コネクタを有する 文書タイプ H E D を使用する
R O C	S P L I T H O U S E からの 入力コネクタを有する R O O への出力コネクタを 有する 文書タイプ H O U を使用する
R O O	R O C からの入力コネクタを 有する S P L I T F L O O R への 出力コネクタを有する ・ ・ ・
F I G. 2 が示すように続く	

F I G. 7 A

【図7】

分割／統合タイプ・セット  
 分割／統合タイプ名  
 分割／統合タイプ属性  
 分割／統合タイプ属性値

**SPLIT HOUSE**  
 分割タイプは「a l l」で  
 ある  
 バンドル分割は「y e s」で  
 ある  
 U T Iからの入力コネクタを  
 有する  
 出力コネクタ・バンドルR O Cを  
 有する

**SPLIT FLOOR**  
 分割タイプは「a l l」で  
 ある  
 バンドル分割は「y e s」で  
 ある  
 R O Oからの入力コネクタを  
 有する  
 出力コネクタ・バンドルC A Rを  
 有する

**SPLIT INSTALL**  
 分割タイプは「a l l」で  
 ある  
 バンドル分割は「n o」で  
 ある  
 C A Rからの入力コネクタを  
 有する  
 出力コネクタセットS A N／H E Aを  
 有する  
 .  
 .  
 .

FIG. 2が示すように続く

FIG. 7B

【図7】

コネクタ・タイプ・セット  
コネクタ・タイプ名  
コネクタ・タイプ属性  
コネクタ・タイプ属性値

プロジェクト開始/UT I  
条件「no」を有する  
先行作業は「no」である

UT I / S P L I T H O U S E  
条件「no」を有する  
先行作業は「no」である

S P L I T H O U S E / R O C  
条件「no」を有する  
先行作業は「no」である

R O C / R O O  
条件「no」を有する  
先行作業は「yes」である

・  
・  
・

FIG. 2が示すように続く

FIG. 7C

【図8】

プロセス・インスタンス・セット			
プロセス・インスタンス名			
家屋建築プロセス			
活動セット	活動名	活動属性	活動属性値
	U T I		プロジェクト開始からの 入力コネクタを有する S P L I T H O U S E への 出力コネクタを有する 文書HEDを使用する 定義状況は「完了」 実行状況「実行済」を 有する
	R O C . 1		S P L I T H O U S E からの 入力コネクタを有する R O O . 1 への出力コネクタを 有する 文書H O U . 1 を使用する 役割「煉瓦職人チーム（J O E）」に 関連する 定義状況は「完了」 実行状況「タスクH1 ROCから開始」を 有する

FIG. 8A

【図8】

ROC. 2  
SPLIT HOUSEからの入力コネクタを  
有する  
ROO. 2への出力コネクタを有する  
文書タイプHOU. 2を使用する  
役割「煉瓦職人チーム」に関連する  
定義状況は「完了」  
実行状況「開始せず」を有する

ROO. 1  
ROC. 1からの入力コネクタを  
有する  
SPLIT FLOOR1への出力コネクタを  
有する  
役割「屋根建築チーム」に関連する  
定義状況は「完了」  
実行状況「開始せず」を有する

ROO. 2  
ROC. 2からの入力コネクタを  
有する  
SPLIT FLOOR2への出力コネクタを  
有する  
役割「屋根建築チーム」に関連する  
定義状況は「開始せず」

・  
・  
・

FIG. 3が示すように続く

FIG. 8B

【図8】

分割／統合セット  
分割／統合名  
分割／統合属性  
分割／統合属性値

SPLIT HOUSE  
分割タイプは「a l l」  
バンドル分割は「y e s」  
UT 1からの入力コネクタを有する  
ROC. 1／ROC. 2への  
出力コネクタを有する

SPLIT FLOOR1  
分割タイプは「a l l」  
バンドル分割は「y e s」  
ROO. 1からの入力コネクタを有する  
CAR. 1への出力コネクタを有する

SPLIT FLOOR2  
分割タイプは「a l l」  
バンドル分割は「y e s」  
ROO. 2からの入力コネクタを有する  
CAR. 2への出力コネクタを有する

FIG. 8C



## 【図8】

SPLIT INSTALL. 1.1  
分割タイプは「all」  
バンドル分割は「no」  
CAR. 1.1からの入力コネクタを  
有する  
出力コネクタセットSAN. 1.1/  
HEA. 1.1を有する

SPLIT INSTALL. 1.2  
分割タイプは「all」  
バンドル分割は「no」  
CAR. 1.2からの入力コネクタを  
有する  
出力コネクタセットSAN. 1.2/  
HEA. 1.2を有する

・  
・  
・

FIG. 3が示すように続く

FIG. 8D

【図8】

コネクタ・セット  
コネクタ名  
コネクタ属性  
コネクタ属性値

プロジェクト開始/UT I  
条件「no」を有する  
先行作業は「no」である

UT I/SPLIT HOUSE  
条件「no」を有する  
先行作業は「no」である

SPLIT HOUSE/ROC. 1  
条件「no」を有する  
先行作業は「no」である

SPLIT HOUSE/ROC. 2  
条件「no」を有する  
先行作業は「no」である

ROC. 1/ROO. 1  
条件「no」を有する  
先行作業は「yes」である

ROC. 2/ROO. 2  
条件「no」を有する  
先行作業は「yes」である

・  
・  
・

FIG. 3が示すように続く

FIG. 8E

【図9】

プロジェクト・セット
プロジェクト名 2軒の家屋を建築する
タスク・セット タスク名 囲まれた活動セット 活動名 タスク属性 タスク属性値  U T I  H 1 R O C 囲まれた活動セット R O C . 1 タスク属性 R O O . 1 にオーバーラップする 予定開始日 XX . XX . XX 予定終了日 YY . YY . YY 割振り労力は30人日 活動状態の役割は 煉瓦職人チーム 割振り済チームは 「煉瓦職人チーム1」 定義状況は「完了」 タスク状況は「開始済」 作業状況は「50%完了」

FIG. 9A

## 【図9】

H1 ROO  
 H1 F1 CAR  
 H1 F1 (SAN & HEA)  
 H1 F1 (IMO & PAI)  
 H1 F2 CAR  
 H1 F2 (SAN & HEA)  
 H1 F2 (IMO & PAI)  
 H1 EMO  
 H2 ROC  
     囲まれた活動セット  
         ROC. 2  
     タスク属性  
     ROO. 2にオーバーラップする  
     予定開始日  
         xx. xx. xx  
     予定終了日  
         yy. yy. yy  
     割振り労力は30人日  
     活動状態の役割は  
         煉瓦職人チーム  
     割振り済チームは  
         「煉瓦職人チーム2」  
     定義状況は「完了」  
     タスク状況は「割振り済」  
     作業状況は「開始せず」  
  
 H2 ROO  
 H2 CAR  
 H2 (SAN & HEA)

FIG. 9B

【図9】

囲まれた活動セット

IMO. 2. 1

IMO. 2. 2

EMO. 2

タスク属性

PAI. 2にオーバーラップする

予定開始日

XX. XX. XX

予定終了日

YY. YY. YY

割振り労力は20人日

活動状態の役割は

塗装工チーム

割振り済チームは

「未割振り」

定義状況は「未完成」

タスク状況は「未割振り」

作業状況は「開始せず」

H2 PAI

H1 USE

H2 USE

FIG. 9C

【図10】

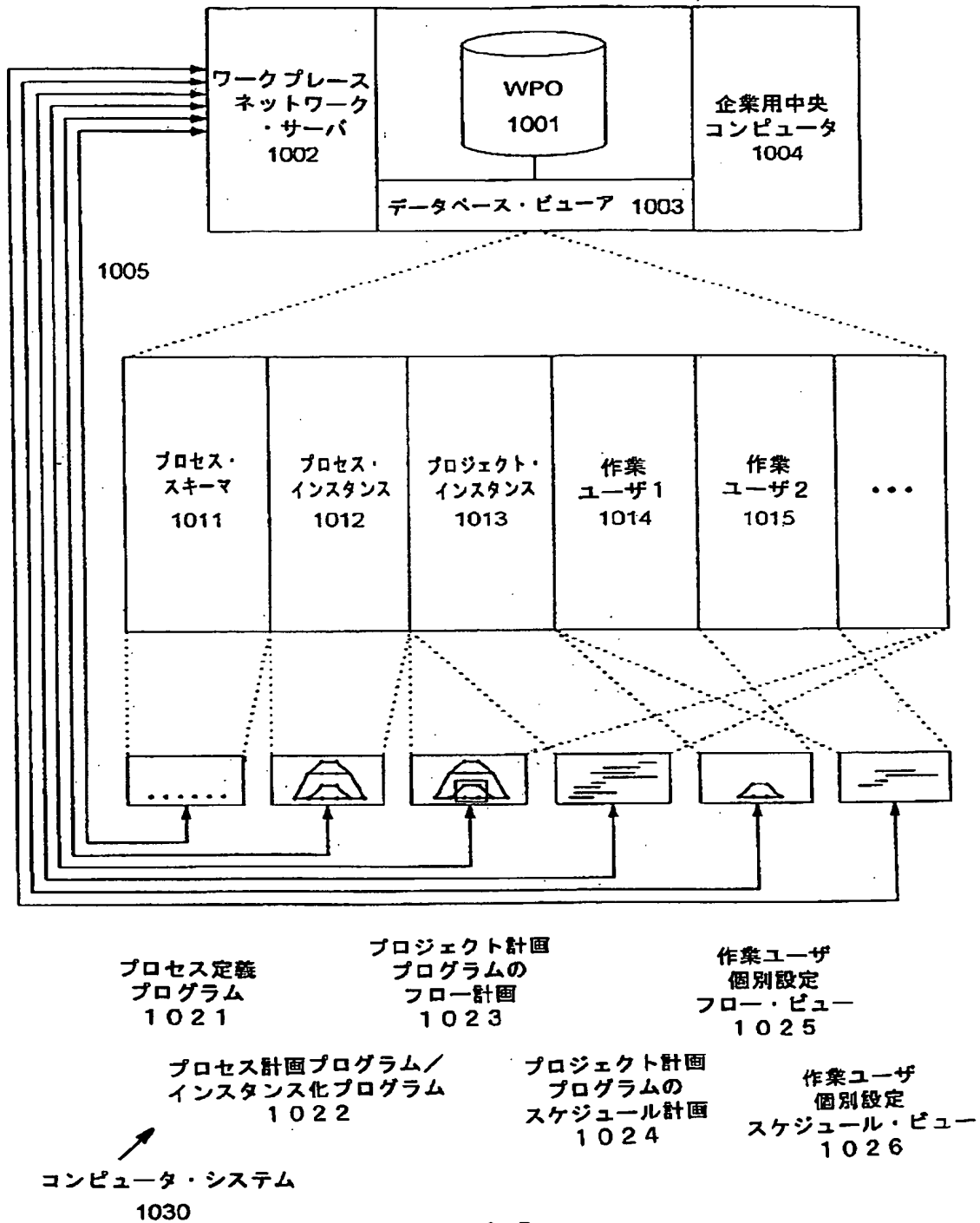


FIG. 10

## 【国際調査報告】

## INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 G06F17/60		International Application No PCT/EP 95/03289
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 6 G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO.A.94 18620 (ACTION TECHNOLOGIES) 18 August 1994 see page 16, paragraph 2 see page 18, paragraph 3 see page 45, last paragraph ---	1
A	WORKGROUP COMPUTING REPORT, vol. 17, no. 5, May 1994, ISSN 1057-8889, US, pages 3-13, XP002001808 R.T. MARSHAK: "IBM's FlowMark. Object-oriented workflow for mission-critical applications" cited in the application see page 3 - page 7 see page 12 --- -/-	1-11
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents : "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to undermand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "&" document member of the same patent family		
Date of the actual completion of the international search  26 April 1996		Date of mailing of the international search report  22.05.96
Name and mailing address of the ISA European Patent Office, P.B. 5814 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016		Authorized officer  Pottiez, M

## INTERNATIONAL SEARCH REPORT

International Application No.  
PCT/EP 95/03289

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>PROCEEDING OF 13TH IEEE ANNUAL INTERNATIONAL PHOENIX CONFERENCE ON COMPUTERS AND COMMUNICATIONS, PHOENIX, US, 12 - 15 April 1994, ISBN 0-7803-1814-5, 1994, NEW YORK, IEEE, US, pages 184-187, XP000462554 L.E. HEINDEL ET AL: "Next Generation Project Management Systems. Part 2: Prototyping" see page 184, right-hand column - page 185, right-hand column ---</p>	1,2,8,9
A	<p>PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS - SYSTEMS ENGINEERING IN THE SERVICE OF HUMANS - LE TOUQUET, FR, 17 - 20 October 1993, ISBN 0-7803-0911-1, IEEE, NEW YORK, US, pages 283-287 vol.4, XP000467583 A. BENLOGAB ET AL: "Hierarchical System Design for Production Planning" see page 283, right-hand column ---</p>	1-3
A	<p>SYSTEMS &amp; COMPUTERS IN JAPAN, vol. 23, no. 11, 1 January 1992, pages 29-40, XP000380835 Y. NAKASHIMA ET AL: "Integrated Expert System with Object-Oriented Database Management System" see page 34 - page 37 ---</p>	1
A	<p>IBM TECHNICAL DISCLOSURE BULLETIN, vol. 35, no. 3, August 1992, NEW YORK, US, pages 4-10, XP000326148 "Transportation Metaphor Workflow Status Display." cited in the application see page 4 - page 7 -----</p>	2



### Information on patent family members

**PCT/EP 95/03289**

Parent document cited in search report	Publication date	Patent family member(s)	Publication date
WO-A-9418620	18-08-94	AU-B- 6133594	29-08-94
		EP-A- 0686282	13-12-95

---

フロントページの続き

- (72) 発明者 ヴェーシュ、エッケハルト  
ドイツ国アイドリゲン、ノイエ・シュタイ  
ゲ 24
- (72) 発明者 ライステン、ウド  
ドイツ国シュットルガルト、マルスヴェー  
ク 6

## 【要約の続き】

記作業プロセス・オブジェクト (WPO) (1001) に報告され、前記作業プロセス・オブジェクト (WPO) (1001) は共通のデータ・ベースとして使用される。本発明の概念によれば、それぞれのビューは特定のワークスペースでサポートし、デジタル・データによって表され、特定の役割に対してその規則を実施しなければならない。作業プロセス情報モデルは、そのプロセスおよびプロジェクト計画モードとその実行の両面で作業プロセスを表すデータ・ベース・オブジェクトの動的定義と使用をサポートする。本発明の概念によれば、1つのオブジェクトは、動的に移動する境界によって区切られたすべてのモードを同時にサポートする。ワークスペースの実施態様では、本発明のシステム内の役割ごとに全体的な規則を実施する。

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**